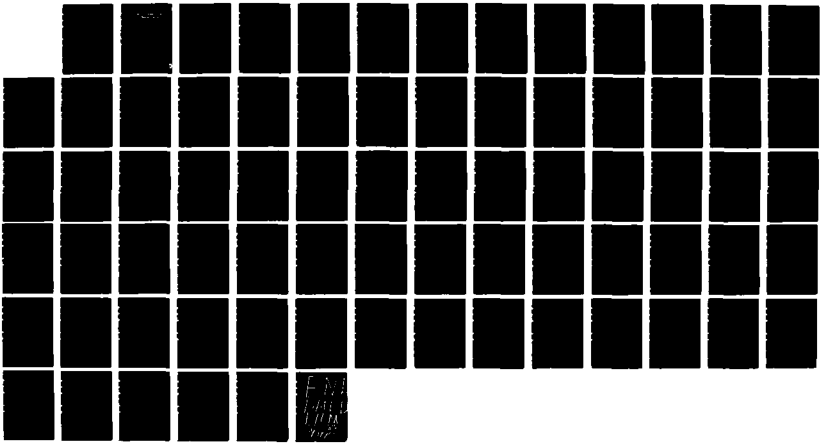
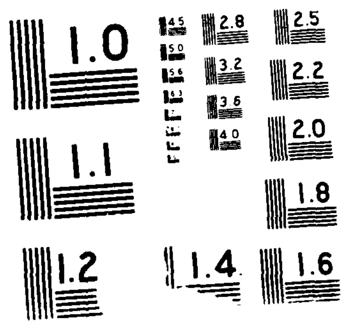
AD-A194 342 ADAPTABILITY AND FEASIBILITY ISSUES CONCERNING THE USE 1/1
OF CD-ROM TECHNOLOGY FOR UNITED STATES NAVY
APPLICATIONS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA
UNCLASSIFIED J S JOHNSON MAR 88 F/G 12/5 NL




1.0
4.5
5.0
5.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.2
1.4
1.6

DTIC FILE COPY

AD-A194 342

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ADAPTABILITY AND FEASIBILITY ISSUES
CONCERNING THE USE OF CD-ROM TECHNOLOGY
FOR UNITED STATES NAVY APPLICATIONS

By

Jimmy S. Johnson

March 1988

Thesis Advisor: Barry A. Frew

Approved for public release; distribution is unlimited

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited
2b DECLASSIFICATION DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) Code 54	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO.

11 TITLE (Include Security Classification)
ADAPTABILITY AND FEASIBILITY ISSUES CONCERNING THE USE OF CD-ROM TECHNOLOGY FOR UNITED STATES NAVY APPLICATIONS

12 PERSONAL AUTHOR(S)
Johnson, Jimmy S.

13a. TYPE OF REPORT	13b TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	1988 March	73

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Paperless ship; CD-ROM Applications for U.S. Navy; Transaction Ledger on Compact Disc (TLOCD)

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The development of CD-ROM technology has produced significant ramifications for mass storage applications. The CD-ROM's read-only nature and its ability to store over 500 megabytes of data on a single disc will eventually revolutionize the historical and archival database industries. The U.S. Navy is particularly interested in the space-saving and weight reduction capabilities of CD-ROM as compared to the current magnetic and paper media. Adaptability and feasibility are the primary issues to be faced when considering the integration of CD-ROM into U.S. Navy applications. This study addresses these issues and determines that CD-ROM will play a significant role in the Navy's efforts to create a "paperless ship" by 1990.

20 DISTRIBUTION AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Barry A. Frew	22b TELEPHONE (Include Area Code) (408) 646-2924	22c OFFICE SYMBOL Code 54Fw

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

U.S. Government Printing Office: 1986—606-24

Approved for public release; distribution is unlimited.

Adaptability and Feasibility Issues Concerning the Use of CD-ROM Technology For United States Navy Applications

by

Jimmy S. Johnson
Lieutenant, United States Naval Reserve
B.S., Middle Tennessee State University, 1979

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 1988

Author: Jimmy S. Johnson

Jimmy S. Johnson

Approved by: Barry Frew

B.A. Frew, Thesis Advisor

Daniel Davis

Daniel Davis, Second Reader

Vincent Y. Lum

Vincent Y. Lum, Chairman,
Department of Computer Science

James M. Fremgen,
Acting Dean of Information and Policy Sciences

ABSTRACT

The development of CD-ROM technology has produced significant ramifications for mass storage applications. The CD-ROM's read-only nature and its ability to store over 500 megabytes of data on a single disc will eventually revolutionize the historical and archival database industries. The U.S. Navy is particularly interested in the space-saving and weight reduction capabilities of CD-ROM as compared to the current magnetic and paper media. Adaptability and feasibility are the primary issues to be faced when considering the integration of CD-ROM into U.S. Navy applications. This study addresses these issues and determines that CD-ROM will play a significant role in the Navy's efforts to create a "paperless ship" by 1990.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. GENERAL REMARKS

CD-ROMs (Compact Disc Read Only Memories) provide computer software applications developers with intriguing possibilities of making hundreds of megabytes, even gigabytes of data readily accessible to personal computer users. Such massive storage capacity opens up new realms of potential applications for microcomputer-software developers.

The CD-ROM has a thousand times the storage capacity of a floppy disk. In the computer industry, we often improve things by a factor of two or three and the new applications are considered evolutionary. But a one thousandfold increase in storage capacity enables us to create rich and multifaceted new applications. (Gates, 1986, p. xi)

Furthermore, a floppy disk can store only a few seconds of full motion, full screen color video, whereas a single CD can store as much as an hour of such video images. The floppy can store only three seconds of high-quality audio, but the CD can store an hour. It is this remarkable power of the CD-ROM disc to digitally store video images, audio, data, and computer code in any combination that emphasizes its vast potential.

CD-ROM technology is derived from CD audio technology and uses the same basic drive mechanisms and disc manufacturing processes. Because of this close relationship, CD-ROM player and disc development has benefitted directly from the technological advances and cost reductions associated with the rapid growth of the CD audio industry. (Einberger, 1987, p. 31)

B. THE TLOCD SYSTEM

Transaction Ledger on Compact Disc (TLOCD) is the culmination of a U.S. Navy supported thesis project conducted in the spring of 1987 at the Naval Postgraduate School in Monterey, California. It involved the transfer of some 2,000,000 records containing historical transaction data from a magnetic tape medium to a CD-ROM disc. The records represented all transactions conducted by the Naval Supply Center at Oakland, California, for the months of October and November 1986. The records were arranged into three types of files according to their particular

application. The "Transaction" files contained data about conducted transactions such as ordering and issuing. The "Closing Balance" files contain such information as quantity on hand and quantity on order. The "Audit Trail" files consist of pertinent data about previous transactions.

Reference Technology Inc. of Boulder, Colorado, was tasked with transferring the data, creating the indexes, and pressing the disc. They also provided the system software to interface between IBM compatible personal computers and the CLASIX Datadrive Series 500 disc player manufactured by Hitachi. A list of the hardware and software initially utilized by the TLOCD system can be found in Table 1.

TABLE 1

TLOCD HARDWARE AND SOFTWARE CONFIGURATION

Zenith Z-248 PC (IBM PC/AT Compatible) with :
-20 Mbyte Winchester Drive
-1 360K Double-sided, double-density
-5 1/4 inch floppy disk drive
-640K RAM
-Intel's 80286 16-bit Microprocessor
-8 MHZ Systems Clock

Zenith RGB/ENHANCED COLOR MONITOR

CLASIX^{tm} DataDrive^{tm} Series 500

SOFTWARE

Standard File Manager
Key Record Manager
Application Specific file access software

Source: Lind Thesis, p. 56.

The evaluation of the TLOCD system attempts to identify an alternative to relieve the over commitment of currently installed TANDEM systems at the eight Naval Supply Centers. The systems are saturated with the Transaction Ledger on Disk (TLOD) database--thus precluding the system from being utilized for more productive

tasks. TLOCD allows the user to query data in much the same way as the TLOD system. The only difference is in the more effective CD-ROM storage medium used by TLOCD. However, the user never actually has to know whether the data is stored by conventional means or whether it resides on a CD-ROM.

C. OBJECTIVES

Unless the file structures for a CD-ROM application are designed carefully, the application's performance is likely to suffer. Typically, poor CD-ROM performance is the result of file-structure design that reflects "magnetic-disk think." Application designers often tend to apply rules of thumb learned from working with magnetic media. Instead, one needs to focus on the unique strengths and weaknesses of the CD-ROM. (Zoellick, 1986, p. 177)

It is the purpose of this paper to examine these strengths and weaknesses in the areas of indexing, file management, and application software issues and to make recommendations to be considered by future Navy research and development in mass storage applications. Additionally, the feasibility and adaptability of CD-ROM technology into U.S. Navy environments will be addressed. The TLOCD prototype will be referenced throughout this report.

II. CD-ROM OVERVIEW

A. GENERAL REMARKS

CD-ROM enjoys tremendous leverage based from the success of digital audio. Both products use the same 12 centimeter plastic disc for storing data, and both employ the same basic manufacturing and playback technologies. CD-ROM thus benefits from the volume-related cost savings that have driven down the prices of digital audio and made it so popular and affordable.

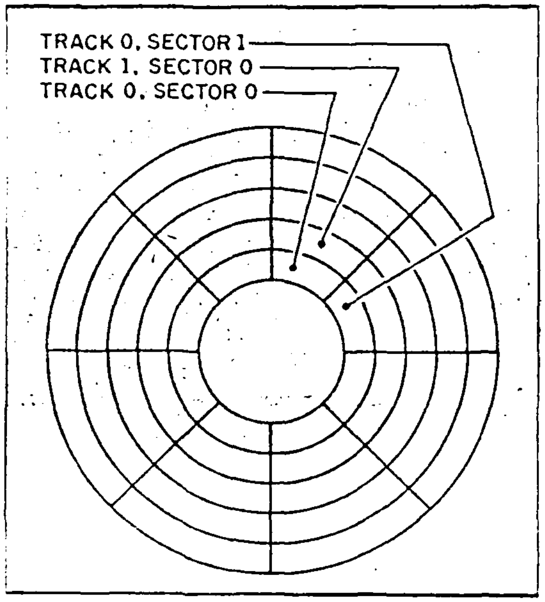
The raw specifications of CD-ROM are staggering. A single 4.72 inch disc stores 550 megabytes of data, the equivalent of 1,500 floppy disks or 28 20-megabyte hard disks. That is 250,000 pages--500 books--whole encyclopedias. Yet any piece of information on the disc can be located and displayed in two or three seconds. (DeTray, 1986, p. 4)

B. PHYSICAL FORMAT

The CD-ROM's physical format is defined by a standard developed by the Philips and Sony corporations and is an extension of their compact digital audio disc standard. However, this digital audio parentage also constrains the CD-ROM to an unimpressive random-seek performance. In particular, the underlying digital audio format results in a data format that is based on constant linear velocity (CLV) recording.

Most magnetic disks use constant angular velocity (CAV) format. Figure 2.1 shows the sector organization of a typical magnetic disk. Note that the sectors on the inner tracks are smaller than those on the outer tracks. This is because CAV is another way of saying constant rotational speed. With a CAV format, the linear velocity of the disk surface relative to the disk head is greater on the outer tracks where the disk's circumference is greater. The outer sectors are also physically larger.

Figure 2.2 illustrates the CLV sector format of a CD-ROM. The relative speed of the disc surface and disc head stays the same, even as the head moves away from the center of the disc. A CD-ROM drive maintains this constant linear velocity by actually changing the disc's rotational speed as the head moves from track to track. The CLV format results in sectors of equal length. The actual number of sectors encountered in a single disc rotation ranges from about nine on the inside of the disc to about 20 on the



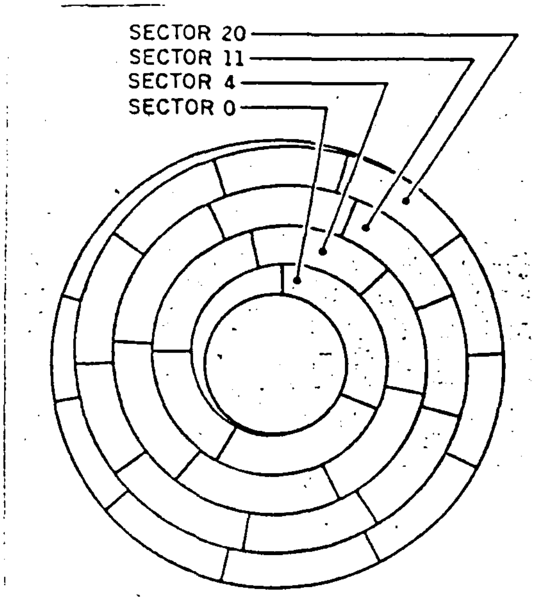
Source: BYTE, May 1986.

Figure 2.1 Sector Organization of a CAV Magnetic Disk.

outer edge. Therefore, recording must be done in a spiral rather than in a series of concentric rings. Recording begins at the inside of the disc and spirals outward.

The great advantage that CAV recording has over the CD-ROM's CLV format is that the CAV organization makes it easier to find the beginning of a particular sector. Suppose one wants to jump to a specific sector relative to the start of a file. With a CAV format, where each track contains a fixed number of sectors, it is very easy to translate this relative sector number into an absolute track and sector address, given the track and sector address of the start of the file.

There is no simple, fixed relationship between a CLV track and the number of sectors on the track. Therefore, translating a relative sector number into an absolute track and sector address is more complicated. In addition, head movement must be accompanied by the mechanical process of speeding up or slowing down the rotational speed of the disc. Together these account for a major part of the CD-ROM's relatively poor performance in locating the desired track. The time required to find the beginning of a particular track is referred to as seek time.



Source: BYTE, May 1986.

Figure 2.2 Sector Organization of a CLV CD-ROM Disc.

On the positive side, CLV recording makes more efficient use of the disc surface. Rather than spreading out data on the outer tracks as on a CAV disk, the CLV format packs the data on the outer tracks just as tightly as on the inner tracks. As a consequence, a CLV disc can hold much more information than a comparably sized CAV disk. From the standpoint of audio recording, where the primary mode of access is sequential, the CLV format is ideal. It packs the maximum amount of music on a disc without exacting a performance penalty. However, when you build a data format on top of this audio format, you pay for increased capacity with decreased seek performance. (Zoellick, 1986, p. 178)

C. PHYSICAL ADDRESSING

The CD-ROM's CLV format rules out using the familiar track and sector addressing schemes used for most magnetic disks. Instead, the CD-ROM uses a scheme that can be traced directly to its audio background. Each disc is said to have 60 "minutes" worth of data. Each minute is composed of 60 seconds and each second is made up of 75 sectors. A single sector can hold 2K bytes of data. Therefore, the entire

disc can hold 540,000K (60 x 60 x 75 x 2K) bytes. The origin of the disc is specified as 0:0:0 (zero minutes, zero seconds, sector zero).

Application developers need not worry about the physical addressing details on CD-ROMs, just as they do not concern themselves with such details on magnetic media. The operating system will convert the physical view into a logical view, allowing the disk to be regarded as a collection of named files rather than a collection of tracks and sectors. Laser-disc operating systems provide the same type of support for CD-ROMS.

D. PERFORMANCE MEASUREMENT

Good CD-ROM software design must reflect an awareness of the CD-ROM's weaknesses, in particular its poor seek performance. Table 2 compares a typical CD-ROM drive with two different types of magnetic-disk drives. The comparisons include capacity, seek performance, and data-streaming performance during a series of sequential reads of contiguous data. The sequential-read performance on the magnetic disk assumes an interleave factor of five, meaning that it takes five disk revolutions to read all the data in a given track.

An average seek on a full CD-ROM takes five times as long as on a 10-megabyte hard disk. When compared to a high-performance magnetic disk, there is more than an order of magnitude of difference in the seek performance. When designing software for a magnetic disk, a major effort to avoid seeks should be made. Given the cost of seeks on a CD-ROM, even more stringent measures should be taken to avoid an average seek. (Zoellic Bill, 1986, p. 180)

However, Table 2 demonstrates that the cost of a short seek covering only a few tracks is relatively small. This is because the CD-ROM only needs to move the mirror used to position the laser beam on the disc. It does not have to move the sled containing the mirror, lenses, and other parts of the disc-reading mechanism. Instead, the laser bounces a pinpoint of light off the CD-ROM's surface, which consists of a pattern of submicroscopic pits. This information is converted into a digital signal and read by an optical disc drive.

This disparity between the cost of a short, local seek and a longer one is of significant importance. It means that every opportunity should be taken to minimize the physical distance between parts of a file to be used in succession. Since the CD-ROM's sequential-read performance as shown in Table 2 is very respectable, reading a large block of data does not cost that much more than reading a short one. The primary cost is in locating or finding the block.

TABLE 2

SEEK TIMES OF CD-ROMS VS. MAGNETIC DISKS

	CD-ROM	Average microcomputer hard disk
Capacity	540 megabytes	10 megabytes
Number of tracks per read head	approximately 18,000	612
Track-to-track seek	1 ms	3 ms
Average seek	500 ms	100 ms
Maximum seek	1 sec	200 ms
Rotational speed	approximately 300 rpm (variable)	3600 rpm
Average latency	100 ms	83 ms
Transfer rate for sequential read	150K bytes/sec	96K bytes/sec

Source: BYTE, May 1986.

E. CD-ROM BENEFITS

The CD-ROM's adequate sequential-read performance and its ability to rapidly seek over the range of a few tracks are important to the design of good software. Its most beneficial characteristic is that it is a read-only medium. It is nonerasable. For applications demanding secure storage of original versions of valuable documents, images, or data streams, the primary advantage of nonerasibility is evident: once the data are recorded, nobody can modify or erase them short of physically destroying the media. (Moore, 1984, p. 72)

Two other benefits arise from the fact that a CD-ROM has a read-only nature. First of all, there are never any concerns with insertions, deletions, or modifications. Therefore, when building a tree, the most frequently used records can be placed in the nodes nearest the roots because they are never going to change. Secondly, the costs of writing and reading are not equally balanced. A CD-ROM is written only once but is read over and over again. Therefore, more time and effort should be put into the initial construction of files and indexes in order to obtain the fastest retrieval possible. Furthermore, building the file and index structures is often done on a larger machine, while the retrieval is most likely to be done on a micro. If expensive tasks such as lexical analysis and text formatting are necessary, it is better to do them once with the

larger computer before creating the disc. Data for a CD-ROM are normally used interactively but are usually prepared in a batch-processing mode. This provides more incentive to do as much work as possible while still in the writing stage. See Table 3 for other CD-ROM advantages.

TABLE 3

ADVANTAGES OF CD-ROM

- **PERMANENT/DURABLE:** It is an excellent archival medium (currently Sony disks are guaranteed for 50 years.) Also very rugged and able to withstand adverse weather and handling conditions.

- **NON-VOLITATILE:** No loss or altering of data during power failure or surges.

- **LOW COST:** The 'per MB' cost of data is less than any storage medium.

- **EXTREMELY PORTABLE:** The media is removable and offers portability of data.

- **SECURITY:** Physical control can be maintained easily and thus large quantities of sensitive data can be controlled. Also, the possiblity exists to manufacture the disk out of glass instead of polycarbonate material and thus, for military purposes emergency destruction could be easily accomplished.

- **SMALL PHYSICAL VOLUME/WEIGHT:** Easily carried, or mailed etc, at a very reasonable expense.

- **NOT ABLE TO BE ALTERED:** This media is Read Only Memory (ROM) and as such, it is extremely useful for audit trails in the legal and financial world where magnetic media have not been allowed as evidence due to the alterability of that media.

- **ENORMOUS DATA STORAGE CAPABILITY:** Up to 600 MB of data on a single side of a single disk which is only 4.72 inches in diameter.

- **USER FAMILIARITY:** It is simply another PC peripheral that, to the user, looks just like a read only MS-DOS etc. disk. Also, the average user has had experience with the same physical disk in the CD-Audio environment and therefore feels more comfortable with it all ready.

- **BACKUP IS ELIMINATED:** There is no need to backup the disk because it is ROM. For safety sake, mulitiple copies can be ordered at the time of disk pressing and stored in separate locations.

- **ELECTRO-MAGNETIC PULSE (EMP) HAS NO EFFECT:** This is not a magnetic media and therefore any sort of electro-magnetic energy has no effect on it.

- **NO HEAD-CRASHES:** The read-device is optical and does not contact the disk in any way, therefore, head-crashes are virtually eliminated.

Source: Lind Thesis, p. 26.

III. CD-ROM APPLICATIONS

A. GENERAL REMARKS

The basic technology for read-only optical discs was developed to distribute movies and high-fidelity music. Consumer electronics companies spent hundreds of millions of dollars over the past decade in Europe, Japan, and the United States to make the videodisc and audiodisc inexpensive, reliable, and long lasting. As a result, data distribution on CD-ROMs was a natural and direct extension of the basic technology. (Hensel, 1986, p. 487)

Information users who have access to a microcomputer and optical disc player are now able to access entire collections of databases that have been placed on CD-ROM. The resulting savings are significant. Even if there is no other reason for buying the microcomputer and disc player, they pay for themselves with a few hours of activity per week when the alternative is online connect charges. However, much greater savings are possible. The Internal Revenue Service has begun a project entitled "File Archival Image Storage and Retrieval" which it estimates will save as much as $36 million annually in storage costs. (Contract, 1986, p. 18)

B. LIBRARY APPLICATIONS

CD-ROM library applications are essentially of two types. On the one hand they are designed as support tools for library automation activities, including traditional book cataloging and local public access catalogs. On the other hand, they provide inexpensive around-the-clock availability of databases previously produced in paper format. (Melin, 1987, p. 509)

A critical problem often faced by librarians is the growth of their collections, especially the periodical and resource indexes. Increasing volumes of new data, in both print and microform, have meant that increased space is needed to house them. The ability of CD-ROM to store hundreds of thousands of pages in a limited space is very appealing for this very reason. The medium is practically indestructible. Not only can dozens of books be stored on disc, but rare and fragile documents, never before made available to the public, can also be stored in their original form without concern that they will be damaged or destroyed by patrons.

Grolier Encyclopedia has already produced a version of the *Academic American Encyclopedia* on optical disc. Also, the Library of Congress is currently conducting a special optical disc pilot program that includes rapid high-resolution scanning, storage and retrieval of images of journal titles, law materials, manuscripts, sheet music, maps, and technical reports. The British Library is experimenting with the development of bibliographic files on CD-ROM.

Moreover, Software Mart, Inc. (SMI) has developed an illustrative dictionary with voice annotation on CD-ROM. It is called *The Visual Dictionary* and could propel illustrated consumer dictionaries into foreign language training vehicles. (Kuhn, 1987, p. 3)

C. MEDICAL AND LEGAL APPLICATIONS

It can be argued that where knowledge is concise, it should be delivered in a concise way. This is particularly applicable to clinical, action-oriented knowledge. (Huntting, 1986, p. 529) Micromedex, Inc. has applied this approach with considerable success and has produced the first medical information product to actually achieve commercial successful distribution with their "Computerized Clinical Information System" (CCIS). The application utilizes highly structured menus that combine easily understood screen displays to bring clinical management protocols into the emergency room with remarkable speed and precision. This design is successful because it recognizes that the emergency room physician or poison center technician is not working in a contemplative environment when he or she has need for the product. On the contrary, there are a multitude of distractions, perhaps even a life hanging in the balance. Consequently, the information must be delivered concisely and accurately with no time for discussion or debate. (Huntting, 1986, p. 531)

The world-wide use of CD-ROM in the medical and health fields continues to grow. The Canadian Center of Occupational Health and Safety has incorporated the largest publicly available chemical database onto a CD-ROM and has included it in its efforts to improve data distribution and employee safety programs. (Abeytunga, 1987, p. 1)

Attorneys and tax accountants must review a tremendous amount of reference material that may be relevant to their clients' legal or tax needs. Equipped with an entire electronic library at their fingertips, attorneys and tax accountants are sure to find it easier to track down and review material and thus improve their ability to serve their clients. CD-ROM is an ideal medium for many legal applications dealing with taxes, statutes, case histories, legal forms, and patents.

D. CARTOGRAPHY APPLICATIONS

One CD-ROM can store a complete digital map of every street in New England plus additional information equivalent to 300 unabridged copies of *Moby Dick*. The basic map information, judiciously compressed, amounts to 120 to 150 bytes per street. Since 60 percent of the U.S. population lives on about one million streets represented in the Census Bureau's files, a simple extrapolation allowing for rural streets that wiggle more than their urban counterparts, yields a nationwide digital map that will fit on a single CD-ROM. (Cooke, 1986, p. 560)

It would be more appropriate to publish regional or state discs supplemented with a wealth of information targeted for specific markets. The business edition, for example, would contain a list of all companies in the region indexed by both industrial classification and geographic location. The family edition would have data about restaurants, tourist attractions, shopping centers, stores, and museums.

DeLorme Mapping Systems of Freeport, Maine, has stored *DeLorme's World Atlas* on CD-ROM. Also, the Compaq Deskpro 386 displays maps of the entire earth from one laser disc in conjunction with a personal computer (Vizachero, 1986, p. 58).

LaserPlot, Inc. has produced the first CD-ROM-based position tracking system for marine navigation. It displays full-color, digitized National Oceanic and Atmospheric Administration (NOAA) charts in various scales (Belanger, 1987, p. 13).

E. U.S. NAVY APPLICATIONS

Current investigation into the interests of CD-ROM technology in the U.S. Navy revealed a NAVSEA sponsored project entitled "Computer-Aided Technical Information System" (CATIS). CATIS is primarily involved with the placing of engineering technical manuals for the Trident-Class submarines onto CD-ROM discs.

Further investigation discovered an ongoing project at the Naval Ship Weapons System Engineering Station (NSWSES) in Port Hueneme, California. The project has been tabbed " Engineering Data Management Information and Control System" (EDMICS) and is involved with placing engineering diagrams onto CD-ROMs for use by major industrial facilities. (Lind, 1987, p. 60)

Image Conversion Technologies has been awarded a $2.5 million contract for image management services for the "Naval Print on Demand" system. ICT will digitize about 1.8 million pages of military specifications to be stored on two 80-gigabyte optical disc library units. ICT's management system will be used for storage, indexing, and retrieval of all documents to be printed, while its order-entry system will be used to

manage orders and perform administrative operations. The anticipated printing volume is 225,000 pages per day with a required turn-around time of two days. (Lind, 1987, p. 61)

The Navy is also conducting research on CD-ROM technology at the Naval Postgraduate School in Monterey, California. The thrust of this research is concerned with the adaptability of systems such as the TLOCD prototype addressed in the introduction of this paper.

IV. THE TYPICAL CD-ROM DATABASE

A. DATA FILES

1. Data Records

The purpose of any database is to provide access to its data records. The data records in a CD-ROM database can be of either fixed length or variable length. The maximum size of a CD-ROM record is 2,147,483,647 bytes, but there must be a memory buffer large enough for the largest record to be read.

2. Data Records and Keys

Keys are fixed-length byte strings which are organized into indexes to provide access to the data records. Keys do not have to be physically contained in the data records and the structure of the records need only be known to the application program. However, if the keys are contained in the records at fixed offsets from their beginning then this information can be stored in the index headers, thus allowing them to be accessed by application programs.

3. Data Records and Indexes

Data record keys are arranged into indexes. Indexing makes it seem that the records of a data file are arranged in the order of the keys for that particular index. Because multiple indexes can be supported, there may be as many orders to the records as there are indexes.

4. Physical and Logical Data Files

Files of data records are provided by the information publisher. For example, the Naval Supply Center in Oakland provided Reference Technology with the data records required for the TLOCD project. The TLOCD application can handle up to 32 files, which is the limit imposed by the Reference Technology file management system. These files can be placed on either optical or magnetic devices or both. All the physical files are logically concatenated to form a single logical data file, and the offsets in the indexes refer to offsets from the beginning of this logical file. A limited update capability can be supported with multiple data files by logically appending new data files to existing data files and creating new indexes for the resulting logical data file. (Key, 1986, p. 17)

B. KEY RECORD FILES

1. Keys

Keys are used to generate key records. Keys may be ASCII character strings, unsigned byte strings with most significant byte first (e.g. left-justified ASCII or EBCIDIC text), signed integers with least significant byte first (e.g. IBM-PC and VAX integers), or unsigned integers with least significant byte first (IBM-PC and VAX unsigned integers).

2. Key Records

Key records are the units from which indexes are formed. They contain a key field, to which other information, including the record's location in the data file, is affixed. Figure 4.1 summarizes the logical structure of key records compiled by Reference Technology.

Key Records fixed length and position up to 32,767 bytes in length The sum of the contents of all key records and data is limited only by the maximum file size of 2 Gbytes-1				
Offset of a data record = 1st 4 bytes, signed integer, LSB* 1st	Data Record Length (optional) if used, 2 or 4 bytes, signed integer, LSB* 1st	Record number (optional) must be used for hash table	Key	Extra Data (optional)

Source: Key Record Manager, p. 18.

Figure 4.1 Key Record Logical Structure.

The data record length is optional because it can be calculated from the offset of the next data record. The key record number is needed only for hash-table indexes, because a record number can be calculated directly from the position of a record in a balanced tree. Duplicate key records are allowed. They are sorted secondarily by data offset in ascending order.

3. Creating Key Record Files

Key record files can be created by the CD-ROM manufacturer or by the data publisher. The decision should be based on the structure of the data records. If the key is in a fixed location in a data record, the key records can be generated automatically by the disc manufacturer. Otherwise, the key records must be provided by the publisher in the format as described in Figure 4.1.

C. INDEX FILES

1. Indexes

Indexes are created by putting sorted key records into an index. Each key index provides access to the data records in the order of the key records that compose it. Key records for an index may be arranged in either ascending or descending order. Each index is assigned an integer identifier, beginning with zero, which is always the data index. Subsequent key indexes are assigned integers beginning with one.

The key records in the data index contain only the byte offsets of the data records in the logical data file. Since the data index is keyed by the record offsets, it provides sequential access to the records in the order they were received by the manufacturer. The data index for databases with records of fixed length is normally a virtual index. For databases with records of variable length, a balanced-tree index containing the record offsets is created. This makes it possible to find a record either by sequential position in the sequence of data records, or by byte offset in the logical data file.

The maximum number of indexes to a Reference Technology database is 2,147,483,647. However, the number of indexes which can be accessed at one time is limited by available memory allocation. Each open index in the database requires memory for an Index Control Block (89 bytes, plus 12 bytes for each level of index) and for a key record buffer. Assuming two-level indexes and 32-byte key records, an IBM PC with 384 Kbytes of available memory could support 2711 open indexes. (Key, 1986, p. 19)

2. Hash Table Indexes

Well-designed hash tables support exact-match key searches with at most one disc access. Positioning by key order will require at most two disc accesses. Partial-match searches are supported, but will require approximately twice as many seeks as the logarithm base two of the number of index pages in the hash table. (Key, 1986, p. 19)

The key records for a hash table are extended to include a key order record number. A cross-reference table is appended to the hash table to allow positioning by key order with the overhead of a single additional disc access, and thereby allowing a binary search of the hash table for partial matches.

3. **Balanced Tree Indexes**

A balanced tree for each index is produced by placing key records in fixed-length index pages, which are arranged in a tree so that examining the records in a page of the tree at one level tells which page to examine at the next lower level. Since there is only one page at the top level, only one page on each level needs to be examined to locate a specified key.

D. CONFIGURATION FILES

A configuration file contains the file specifications (the complete volume, path, and name) of each of the data files and index files that make up a database. Its function is to map the logical correspondences between index identifiers and the physical indexes. Performance considerations may request certain index files to be copied to a magnetic device. For this reason, a configuration file contains only printable ASCII characters. This allows the use of a text editor to modify the volumes or paths in a magnetic copy of a configuration file. (Key, 1986, p. 24)

V. KEY RECORD UTILIZATION

A. KEY RECORD MANAGER

Key Record Manager is a software access program for files with structured fields and records. It was designed by Reference Technology primarily as a tool to be used in conjunction with CD-ROM databases. It provides an Indexed Sequential Access Method (ISAM) comparable to mainframe retrieval systems for record-oriented databases. The *Key Record Manager* allows for two index structures, a balanced tree and a hash table. The *Key Record Manager* software is implemented as a library of C language functions that can be linked to application programs which require access to supported databases.

B. SAMPLE DATABASE

CD-ROM databases normally consist of large files, each organized into similarly structured data records which are divided into fields. The data record fields consist of key fields which are indexed and data fields which are not. The easiest way to conceptualize such a database is in two dimensions. A data record, the individual entry for a database, is the row; the field is part of a column of similar information for each of the rows.

Figure 5.1 is an example of a simplified, fictitious stock market database. It was reproduced from Reference Technology's *Key Record Manager* and will be referred to throughout the remainder of this chapter. The data records in this example are of variable length and are arranged in the alphabetical order of their ticker tape symbols.

The offset field refers to the offset of the record from the beginning of the data file. It is not usually represented within the record but is implicit in the ordering of the records within the file. The comment field is text which is not shown completely because it varies in length for each company.

C. USING KEYS TO BUILD KEY RECORDS

There must be a sorted file of key records in order to construct indexes. It should be placed in a hash table or tree for quick access. The key fields of the records are used to create key records which contain a copy of the key field and the offset of the record associated with that particular key field in the data file. Figure 5.2 shows a key record generated from the Dividend field in one of the data records.

Offset	Symbol	Name	Exc.	SIC	Price	Earnings	Div.	Date	Comment
0	ABC	AgBusCo	N	0112	22	3.50	1.60	3/31/86	Corp. farming is
10722	BAC	Tobacco	A	0144	15	(.71)			Chewing tobacco
21677	CAB	Taxico	N	4577	42	6.57	.45	1/15/86	This taxi compan
29941	CAR	Meat,Inc	O	0128	8	.45	.72	5/1/86	Meat products for
40443	DDE	Dicers	O	5770	17	1.21			This fast-growin
56678	DRI	Realest	N	6344	1	(3.44)			Poor investments
73419	DST	DenStand	O	5057	34	1.21			Designer jeans, i
88007	EBR	EBanks	O	6776	34	5.22	1.60	3/1/86	Regional banking
101571	EST	Clocks	N	5470	22	2.11	.72	4/1/86	Despite its name
110849	FIN	Finbank	O	6776	13	1.86	1.00	1/15/86	Suspending the di

Source: Key Record Manager, p. 6.

Figure 5.1 Sample Stock Market Database.

D. USING KEY RECORDS TO CREATE INDEXES

The indexes can be constructed once all the key records have been created from the database keys. A complete database would contain indexes for all the data record keys. The indexes are in turn placed in index files and are used to access the data records themselves. The indexes could all be placed in one file or they could be placed in separate files. Figure 5.3 contains all the indexes generated for the key fields in the sample database. Note that some of the fields such as Exchange, Date, and Comment are not key fields and therefore cannot be searched.

E. SEARCHING INDEXES

Indexes are a space-saving device because they are made up of key records rather than whole data records. Only one set of data records need be mastered onto a CD-ROM disc, with access to the single copy of the data records being made available in a different order depending on which index is utilized. This requires much less space than putting the data records on the disc in different places for different sort sequences.

The data records on a CD-ROM have the sequence shown by their offsets and will always retain that order in the data file. However, the indexes to the data records

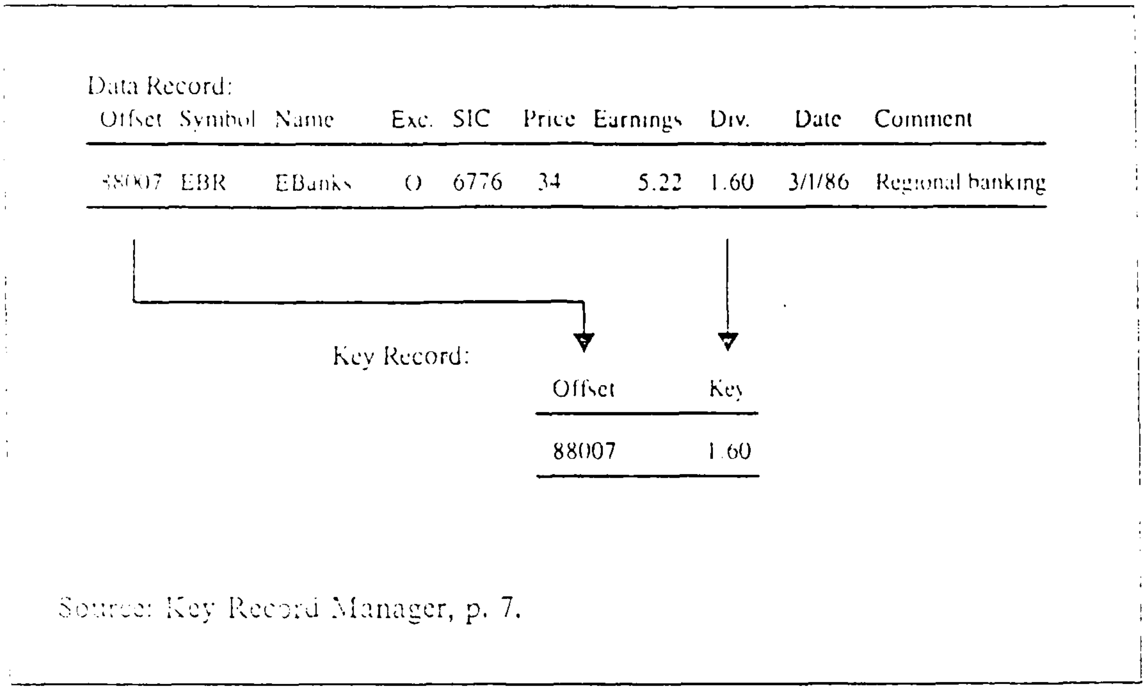


Figure 5.2 Key Record Generation.

have the order of their keys which have previously been sorted. Therefore, creating indexes for the key fields makes it seem as if the data records are arranged in a series of different orders, one for each index used to access them. In our example, the data index (Index 0) is used to access the records in their original order. Figure 5.4 shows the order of the records when indexed by Name (Index 2) and when indexed by Price (Index 4).

Conceptually, the search for a matching key is accomplished by beginning at one end of the key sequence and searching the keys sequentially towards the other end until a match or close match is found. For ascending searches, the first key equal to or greater than the desired key will be retrieved. For descending searches, the first key equal to or less than the desired key will be retrieved. Thus one could search the Name index for "Tob" and retrieve "Tobacco" if the search is ascending, or retrieve "Texaco" if the search is descending. In reality it is not a sequential search but is actually a balanced tree traversal or hash table look-up. Care should always be taken to design these structures so that the number of comparisons and accesses can be minimized.

Index 0 (Data index)	Index 1 Offset	Index 1 Symbol	Index 2 Offset	Index 2 Name	Index 3 Offset	Index 3 Industry Code
0	0	ABC	0	AgBusCo	0	0112
10322	10322	BAC	101571	Clocks	29941	0128
21677	21677	CAB	73419	DenStand	10322	0144
29941	29941	CAR	40443	Dicers	21677	4577
40443	40443	DDE	88007	EBanks	73419	5057
56678	56678	DRI	110849	Finbank	101571	5470
73419	73419	DST	29941	Meat, Inc	40443	5770
88007	88007	EBR	56678	Realest	56678	6344
101571	101571	EST	21677	Taxico	88007	6776
110849	110849	FIN	10322	Tobacco	110849	6776

Index 4 Offset	Index 4 Price	Index 5 Offset	Index 5 Earnings	Index 6 Offset	Index 6 Dividend
56678	1	56678	(3.44)	10322	
29941	8	10322	(.71)	40443	
110849	13	29941	.45	56678	
10322	15	40443	1.21	73419	
40443	17	73419	1.21	21677	.45
0	22	110849	1.86	29941	.72
101571	22	101571	2.11	101571	.72
73419	34	0	3.50	110849	1.00
88007	34	88007	5.22	0	1.60
21677	42	21677	6.57	88007	1.60

Source: Key Record Manager, p. 8.

Figure 5.3 Key Created Indexes.

The records in the example, when accessed by Name (Index 2) would appear to be ordered as follows:

Offset	Symbol	Name	Exc	SIC	Price	Earnings	Div.	Date	Comment
0	ABC	AgBusCo	N	0112	22	3.50	1.60	3/31/86	Corp. farming is
101571	EST	Clocks	N	5470	22	2.11	.72	4/1/86	Despite its name
73419	DST	DenStand	O	5057	34	1.21			Designer jeans, i
40443	DDE	Dicers	O	5770	17	1.21			This fast-growin
88007	EBR	EBanks	O	6776	34	5.22	1.60	3/1/86	Regional banking
110849	FIN	Finbank	O	6776	13	1.86	1.00	1/15/86	Suspending the di
29941	CAR	Meat,Inc	O	0128	8	.45	.72	5/1/86	Meat products for
56678	DRI	Realest	N	6344	1	(3.44)			Poor investments
21677	CAB	Taxico	N	4577	42	6.57	.45	1/15/86	This taxi compan
10322	BAC	Tobacco	A	0144	15	(.71)			Chewing tobacco

If accessed by Price (Index 4), the apparent order of the data records would be:

Offset	Symbol	Name	Exc.	SIC	Price	Earnings	Div	Date	Comment
56678	DRI	Realest	N	6344	1	(3.44)			Poor investments
29941	CAR	Meat,Inc	O	0128	8	.45	.72	5/1/86	Meat products for
110849	FIN	Finbank	O	6776	13	1.86	1.00	1/15/86	Suspending the di
10322	BAC	Tobacco	A	0144	15	(.71)			Chewing tobacco
40443	DDE	Dicers	O	5770	17	1.21			This fast-growin
0	ABC	AgBusCo	N	0112	22	3.50	1.60	3/31/86	Corp. farming is
101571	EST	Clocks	N	5470	22	2.11	.72	4/1/86	Despite its name
73419	DST	DenStand	O	5057	34	1.21			Designer jeans, i
88007	EBR	EBanks	O	6776	34	5.22	1.60	3/1/86	Regional banking
21677	CAB	Taxico	N	4577	42	6.57	.45	1/15/86	This taxi compan

Source: Key Record Manager, pp. 9-10.

Figure 5.4 Searching On Specified Indexes.

F. KEY RECORDS FOR SPECIAL PURPOSES

1. Partial Keying of Data Records

Index performance is generally better when smaller key records are involved. This is especially true for balanced trees where key records may result in additional tree levels and therefore cause additional disc accesses. Index size can be greatly reduced in some cases if some data records are not keyed on every index. Since the Symbol index in our example database is in the same order as the data records it becomes possible to key only the first record in each CD-ROM sector. Then a partial match search in the much smaller resulting index could be followed with an exact match search in the data records themselves. Index size can also be reduced by not indexing records on key fields that are blank.

2. Key Records With Extra Information

Key records may contain additional information besides the key and offset fields. Figure 5.5 displays such a record. A length field may be included for variable-length records. However, it is not essential because the length of the data record could be determined by finding the offset of the next data record and subtracting, but this would require an extra access to the data index (Index 0).

Data Record:

Offset	Symbol	Name	Exc	SIC	Price	Earnings	Div.	Date	Comment
73419	DST	DenStand	0	5057	34	1.21			Designer jeans, i

↓

Key Record:

Offset	Record length (optional)	Key Number (hashing only)	Key
73419	14588	6	DST

Source: Key Record Manager, p. 12.

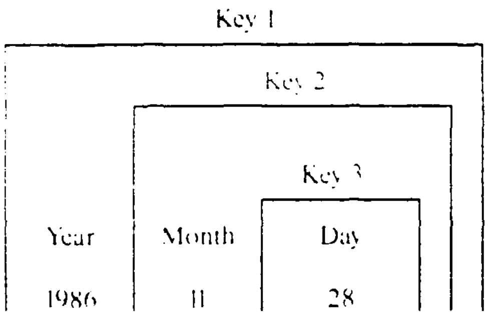
Figure 5.5 Keys with Additional Data.

If hash tables are used, a key number is required because the record entries in a hash table are not arranged by the order of their keys. Hash table keys are distributed randomly across index pages and are only sorted within a page. The keys in a balanced-tree are arranged in a fully sorted pattern and therefore do not need a key number.

One option which can affect application performance and disc overhead is that key records can also contain extra or optional data for use only by the application program. Once a key record is located within an index, the optional data can be read immediately from the key record and thus save an access to the data file. Appending extra data to keys makes retrieval of that data very quick, once the key is located. This is obtained at the expense of a larger index which would require a longer seek. However, a second seek to locate the additional data is no longer necessary.

3. Overlapping Keys

Another area in which key record design can affect application performance is the overlapping of key fields by other key fields. For example, it might be desirable to allow a date field (Year-Month-Day) to be searchable by various overlapping keys as seen in Figure 5.6. This overlapped set of keys could be used to search on Year-Month-Day (Key 1), Month-Day (Key 2), and Day (Key 3) information. By searching for partial matches Key 1 could also be used to search on Year-Month or Year, and Key 2 could be used to search on Month. The same searches could be performed with separate Year, Month, and Day fields, but this would mean searching in three separate indexes for a Year-Month-Day specification, with much worse than triple the access time for this index. (Key, 1986, p. 13)



Source: Key Record Manager, p. 13.

Figure 5.6 Overlapping Keys.

VI. CD-ROM INDEXING STRATEGIES

A. BALANCED-TREE INDEXES

1. Tree Construction

The general form of a tree structure on a CD-ROM is similar to that of a broad, shallow balanced-tree. Since CD ROMs are not concerned with insertions and deletions the blocks of the tree can be packed completely full. This results in the tree using less space and in each block having a larger number of children. Moreover, a broader, shallower tree is produced.

If balanced-trees are built by inserting records randomly and if procedures developed for handling the growth of dynamic trees are used, the blocks of the tree will be between 50 and 100 percent full with an average utilization of between 67 and 85 percent (Zoellick, 1986, p. 184). That is, trees will contain blocks that are not completely full. A special tree-loading procedure that does not use the normal block-splitting method involved in balanced-tree insertion is needed.

The first step in developing an appropriate tree-loading procedure is to sort all the records by their keys as discussed in Chapter Five. The sorted records are then written one at a time into the leftmost block at the lowest level of the tree. When that block is full it is written out to disc. The next record goes into a parent block. Then the next block at leaf level is filled. When this second leaf block is full, it is written out to disc and another single record is placed in the parent block. This process continues until all the records have been loaded. Figure 6.1 shows that all the records are arranged in the blocks in a numbered sequence.

The primary advantage of this loading procedure is that it capitalizes on the read-only nature of the CD-ROM by building a shallow tree and avoiding seeks. There is also an important second advantage. If each block is written out as soon as it is full, then parent blocks will be stored in close proximity to their children, making use of the CD-ROM's better performance on short, local seeks. Furthermore, the proximity of parents and children will never be threatened since the balanced-trees used for CD-ROM are not dynamic.

There are other possibilities for decreasing seeks if something is known about the distribution of requests for the records stored in the tree. Say, for example, that it

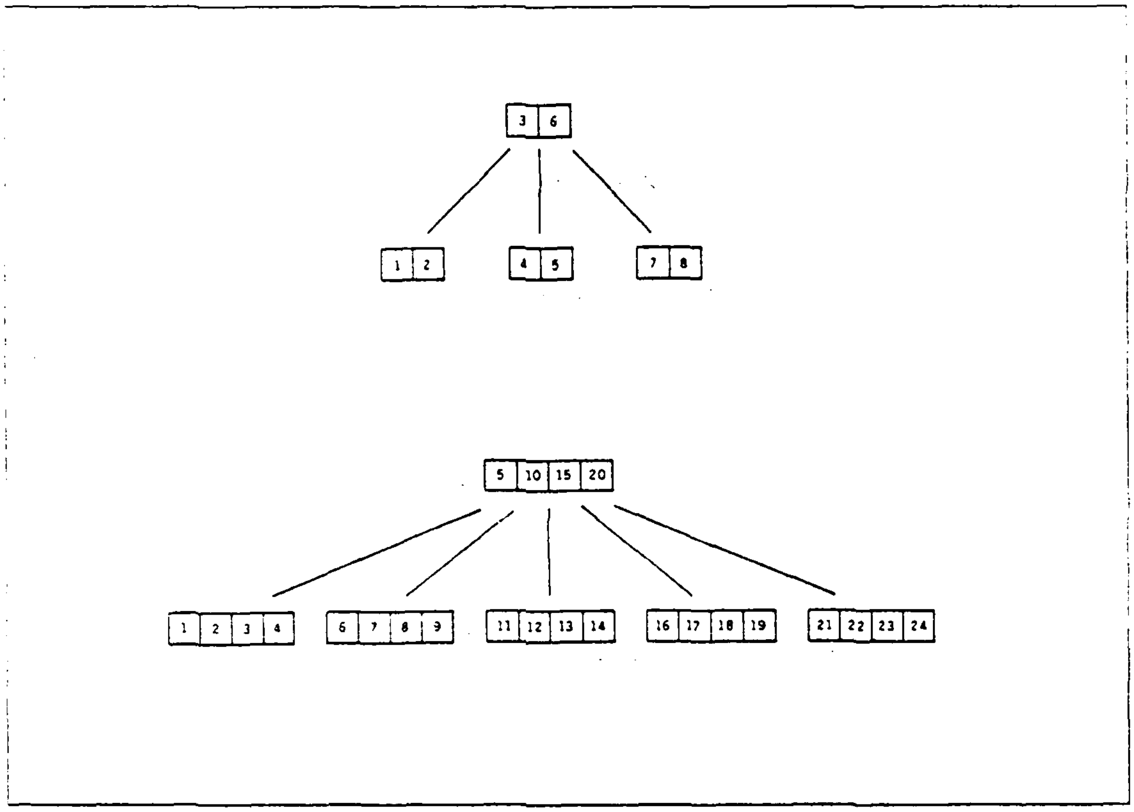


Figure 6.1 Properly Loaded Balanced-Trees.

is known that 85 percent of the requests are for 10 percent of the records. The number of seeks can be greatly reduced if the tree-loading procedure can be designed to place the most frequently used records as near the root as possible.

2. **Tree Optimization Formulas**

The following formulas were used by Reference Technology in designing the *TLOCD* database:

- $L >= \log(N + 1) / \log(P + 1)$ L is the # of tree levels
- $P >= \sqrt[L]{N + 1} - 1$ P is # of key records in an index page
- $N <= (P + 1)^L - 1$ N is # of key records in the index

These formulas relate number of key records, number of tree levels, and page size and are used to optimize balanced-tree performance for CD-ROM databases. Table 4 displays examples of how the formulas can be used.

TABLE 4

OPTIMIZING BALANCED-TREE PERFORMANCE

Given the number of records, page size, and key record size, the minimum number of tree levels can be calculated:

Number of key records = 100,000,000
Page size = 4096 bytes
Key record size = 8 bytes

$$N + 1 = 100{,}000{,}001$$
$$P + 1 = (4096 / 8) + 1 = 513$$
$$L \geq \log(N + 1) / \log(P + 1) = 2.95$$

Since there must be an integral number of levels, 3 levels are required.

Given the number of tree levels, number of records, and record size, the minimum page size can be calculated:

Number of tree levels = 2
Number of key records = 2,000,000
Key record size = 32 bytes

$$N + 1 = 2{,}000{,}001$$
$$1 / L = 1 / 2 = .5$$
$$P \geq ((N + 1)^{(1/L)}) - 1 = 1413.21$$

Since there must be an integral number of records on a page, the page size must be large enough for 1414 records. If the page size is divisible by 2048 bytes (the CD-ROM sector size) a 47,104-byte page size is needed.

Given the number of levels, page size, and key record size, the maximum number of records can be determined:

Number of tree levels = 2
Page size = 4096 bytes
Key record size = 8 bytes

$$L = 2$$
$$P + 1 = (4096 / 8) + 1 = 513$$
$$N \leq ((P + 1)^L) - 1 = 263{,}168$$

At most 263,168 records can be placed in this tree.

Source: Key Record Manager, pp. 21-22.

B. HASHED INDEXES

1. Overflow Avoidance

Hashing fits the strengths and weaknesses of the CD-ROM perfectly for applications that do not need to access records in order by key. It consists of using a function to transform each record's key into a bucket address within the file. In order to find a particular record, the function is applied to that record's key, and then retrieves the bucket at the resulting address. Hashing works well and permits single-seek retrievals as long as long as there is room for each record in its associated bucket. The following variables can be manipulated to guarantee that overflow does not happen:

- packing density of the hashed storage
- the size of the bucket
- the design of the hash function

Packing density and bucket size are discussed further in the next chapter.

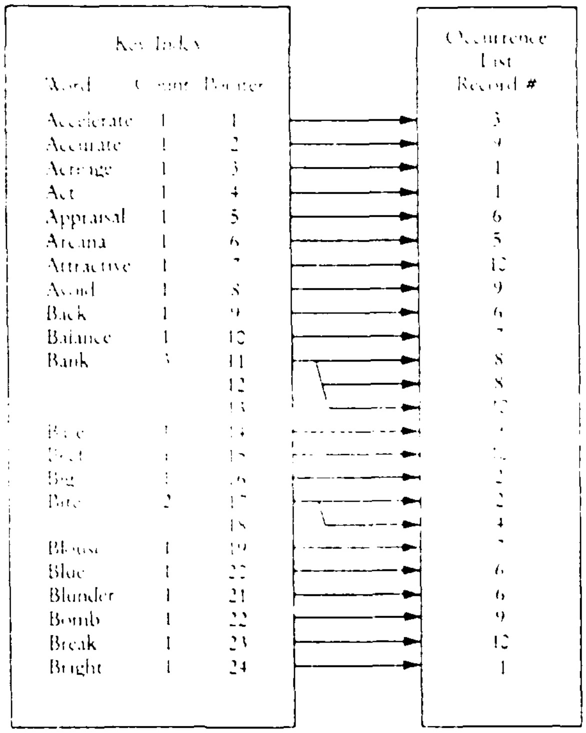
2. Hashing Functions

Since CD-ROM is a read only medium, there exists a complete list of the keys to be hashed before the file is built. The keys can be analyzed to discover functions that would distribute them more uniformly than a random function would. A perfectly uniform distribution would place an equal number of records in each bucket and guarantee no overflow even at a packing density of 100 percent. Although developing such a function can be very time-consuming, an economical way of improving on purely random distributions can often be found.

The CD-ROM's read-only nature makes it possible to optimize a hash function. It is also practical because large computers operating in a batch mode can be used to create the data set that will be used interactively by small computers.

C. INVERTED INDEXES

Inverted files are ideally suited for full-text fields because when used with structured fields containing repeating key values they save index space. A copy of each key value is stored in an index along with a pointer to a list of all records associated with the key. The Comments field in applicable databases is normally a full-text field and a good candidate for an inverted index. If each word is used as a key in a key record, the same words will occur over and over again and create a very large index. An inverted file stores each word only once to represent all of its occurrences and results in a much smaller index. Figure 6.2 represents an inverted index for words beginning with 'A' and 'B' from a fictitious database.



Source: CD ROM Optical Publishing, p. 115.

Figure 6.2 Inverted Indexing.

Such sophisticated indexing schemes can sometimes require as much or more space as the data itself. The *Grolier Electronic Encyclopedia* requires 60 megabytes to accommodate the text and 50 megabytes to accommodate the indexing. (Dixon, 1987, pp. 10-17)

D. CHOOSING THE PROPER INDEX STRUCTURE

Because CD-ROM discs are a read-only medium, the choice of index structure must be made when the database is designed. It is possible to use more than one type of index on a single database so that it becomes feasible to choose whichever type offers the best performance for individual key fields.

Balanced tree searches are best for applications where partial-match searches are frequent. This is because the index can be ordered by the key value. They also perform well in exact-match applications when it is desirable to minimize the index size. Balanced-trees used in CD-ROM applications waste no space and can typically acquire any key in the index with only two or three accesses.

Hash tables perform best when the quick access of exact matches is the main consideration. Normally, hash tables can be constructed so that only one disc access is sufficient. However, hash-table indexes are not as compact as balanced-trees and will typically be 20 to 50 percent larger than a comparable balanced-tree index. Furthermore, hash tables perform partial-match searches poorly because it is nearly the same as searching a sequential file. (Colvin, 1987, p. 115)

Boolean and relational operations on CD-ROM discs are best supported by inverted files. Either hash tables or balanced trees can be used to create the files. Since all data record numbers containing a particular key value are listed together in an inverted file, it must be loaded into a rather large memory buffer to minimize accesses to the CD-ROM.

The index structure used in the development of TLOCD was a combination of a balanced-tree and a hash table. In this way the time required to perform both partial and exact-match searches could be minimized.

VII. CD-ROM FILE MANAGEMENT

A. GENERAL DIRECTORY STRUCTURE

The High Sierra Standard entails a hierarchical structure of descending subdirectories branching down from the parent directory. This directory structure is called a "Standard File Structure," and there must be only one per CD-ROM disc. A path table operates as an index to each subdirectory and provides a pointer to the logical block number where the subdirectory is located. A path table obviates the need to sort each level of the directory hierarchy in the search through the directory structure. Under certain circumstances, the path table can be contained in RAM, providing one-seek access to the subdirectory of interest. This occurs when the subdirectory names are short enough and the number of subdirectories small enough so that the path table can reside in one physical logical sector. (Approximately 128 subdirectory names of eight characters each will cause the path table size to be about 2048 bytes or one logical sector.) Thus, given an eight-level tree, holding a path table in RAM saves seven seeks. (Standard, 1986, p. 2.4)

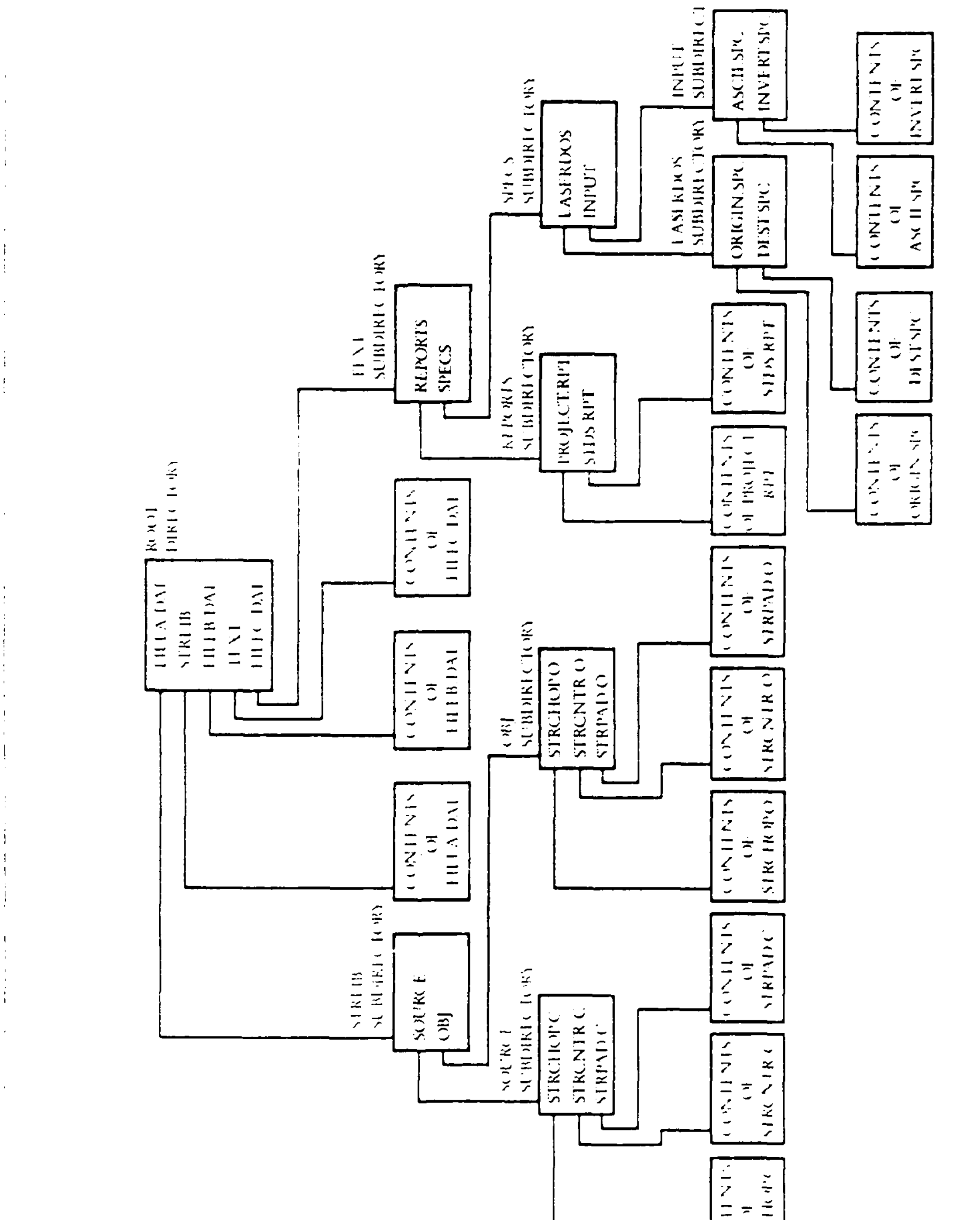
B. DIRECTORY STRUCTURE DESIGN

1. Multiple-File Explicit Hierarchies

This type of directory structure is used by UNIX, MS-DOS, VMS, and other magnetic disk systems. Early versions of Digital Equipment's UNIFILE system are an example of a CD-ROM file system that used this kind of directory structure. This particular structure as shown in figure 7.1 allows subdirectories to be treated as files. It is an excellent system for magnetic disks because it provides the flexibility required in order to add new subdirectories and delete old ones. However CD-ROMs do not require such flexibility. Furthermore, we cannot afford the time to seek from subdirectory file to subdirectory file in order to find a file with a long path name such as:

johnson programs source acctg ledger post.c

The strong features of this type of directory structure are familiarity and the fact that it handles generic searches reasonably well. Moreover, by taking advantage of the CD-ROM's read-only nature, the files in each subdirectory can be sorted and improve generic searching even more.



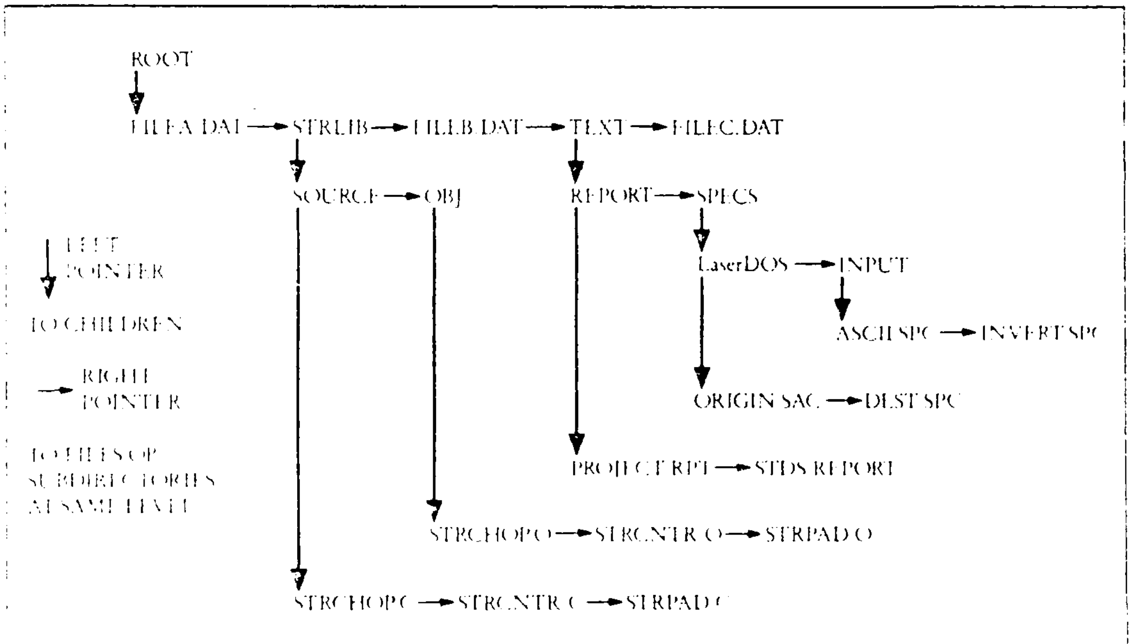
Source:
CD-ROM The New Papyrus, p. 124.

Figure 7.1 Multiple-File Explicit Hierarchy.

The main disadvantage is that we must search through an entire level of the directory structure while looking for a file. If all the files are in the root then a search for a single file would involve the whole directory. Even if the files are sorted within each directory level, a binary search of a large single-level directory containing 10,000 files would require a dozen or more seeks back and forth across the sectors that make up the directory.

2. **Single-File Explicit Hierarchies**

This approach to directory hierarchies involves placing the entire directory structure in a single file. The root directory and all subdirectories are treated as records within a file rather than separate files. Figure 7.2 represents this type of structure, which was used in the first version of LaserDos, a tree-oriented system designed by TMS, Inc. for optical discs. The left pointers from the subdirectory records point to elements in the subdirectory. Right pointers always point to files or subdirectories at the same level.



Source: CD-ROM The New Papyrus, p. 116.

Figure 7.2 Single- File Explicit Hierarchy.

The important benefit realized from compressing the directory hierarchy into a single file, rather than spreading it out by using a different file for each subdirectory, is that we can often cut down on the number of seeks required to open a file. A somewhat small directory containing no more than two hundred files can be contained in just two or three sectors which could easily fit in RAM. This holds true even if there are many levels of subdirectories. Therefore, the single-file explicit hierarchy can often improve on the performance of multiple-file explicit hierarchies when opening files that have path names containing several subdirectory levels.

3. **Hashed Directories**

Any file can be opened in one seek if we hash the entire path and file name to an address within the directory. This will work even if there are tens of thousands of files on the disc. A hash function would transform the character string representing the path and file name into the address of a hash bucket. A seek to the directory bucket would gain access to the information needed to open a file.

If the hash buckets can be prevented from overflowing, then it can be guaranteed that the hashing procedure would require no more than a single seek. If overflow occurred, one or more seeks would be required in order to locate the information that had to be stored elsewhere. The read-only nature of the CD-ROM makes it possible to manipulate the packing density of the directory file. Overflow can be avoided by placing a small number of records into a large file. The more tightly a file is packed, the more likely it is that at least one bucket will overflow. The bucket size also affects overflow. No overflow could be guaranteed if the entire file was considered to be a single bucket. Unfortunately, the entire file would have to be read into and processed in RAM.

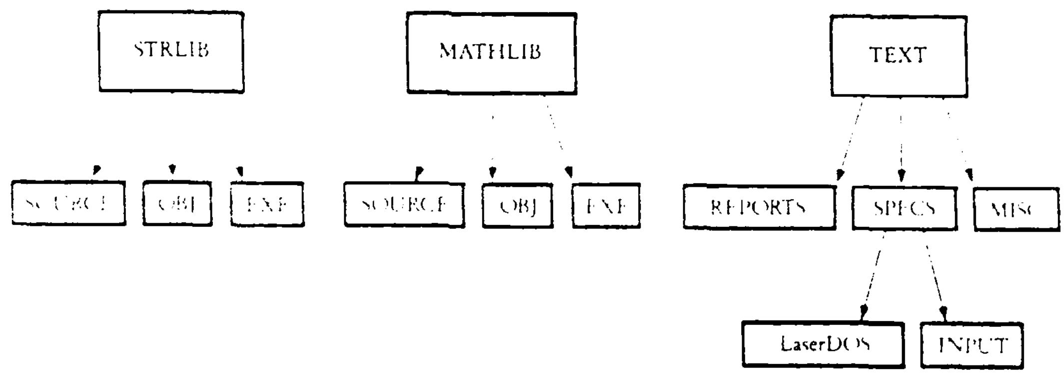
4. **Indexed Directories**

The key to the success of this approach is a structure called a path table. The path table provides a compact mechanism for quick translation of the full path for a subdirectory into an integer called the path identifier. The path identifier is actually the relative position of each file obtained from a level order traversal of the directory hierarchy. By examining Figure 7.3 the path identifiers for the following path names can be determined:

- strlib = 1
- mathlib = 2
- text = 3
- strlib/obj = 5

- mathlib source = 7
- text reports = 10
- text specs input = 14

The path table's ability to compress an entire directory path into a two-byte integer guarantees that directory records can be kept relatively short and that many directory records can be put into each block of the directory structure.



Source: CD-ROM The New Papyrus, p. 119.

Figure 7.3 Index Path File Structure.

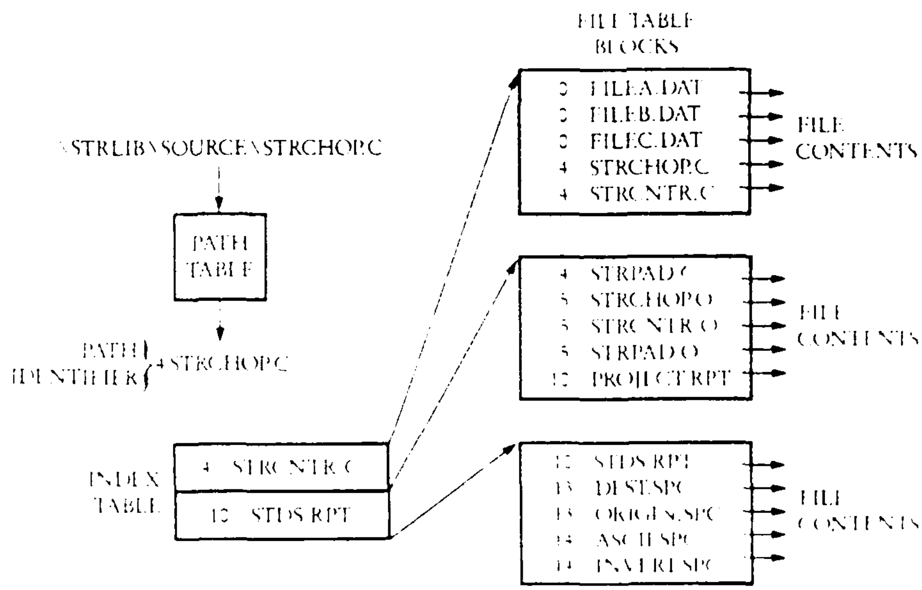
After performing an average seek of about .5 seconds, a minimum of one two-byte sector is read in from the disc. For an additional cost of six milliseconds another 6K bytes can be read in, making a total of 8K bytes in all. If the size of the directory records can be held to 32 bytes each, then each seek out to the CD ROM can bring in as many as 256 records for an 8K block.

The file records are placed into the blocks of a file table which contain the information needed to open any file in the file system. They are arranged according to their path identifier which was extracted from the path table. As a result, all the files in a single subdirectory are grouped together (i.e., they have the same path identifier) and then ordered by name. This structure supports efficient generic and binary searching.

When a particular file is to be opened, we need to find the block in the file table that has the record corresponding to the desired path identifier and file name. The costly part of the file search is the seek to the block's beginning, so it is desired to find the right block on the first attempt. To ensure this occurs, an index table is used to tell the path and file names that are at the block boundaries. Figure 7.4 displays an overview of the contents in the file table. Now suppose the file to be opened is:

\strlib\source\strchop.c

It is shown in Figure 7.4 that the request starts at the path table and converts the path name into a path identifier of '4'. The index table is then searched for "4strchop.c". Since the value of "4strchop.c" is less than the first entry (alphabetically), it follows the first pointer from the index table to find the first block in the file table where it finds the location of the file and other information required to open it.



Source: CD-ROM The New Papyrus, p. 122.

Figure 7.4 Using an Indexed Path Directory.

The path table's compression ability allows for short directory records so that many of them can be packed into each block of the file table. This reduces the total number of blocks required for the file table. A small file table will result in a small index table. It would be very desirable to store both the index and path tables in RAM

rather than forcing a disc seek every time we needed them. In this way the indexed directory allows the opening of any file with only one seek to the CD-ROM.

C. BLOCKS AND BUFFERS

1. Determining Block Size

A general rule applying to any file structure design is to make each disk seek as profitable as possible. This is the reason why paged structures such as balanced-trees are commonly utilized. Each access to the disc retrieves enough data to make decisions about the next tree level instead of making a simple two-way choice in a binary tree. The disc is never accessed to retrieve only one record but to retrieve a block of records that can be read and processed much faster in RAM. Even though CD-ROM seeks slowly, it can acquire a large block of data at an acceptable rate. Therefore, the choice of the block size is extremely important.

Both physical and logical design factors should be considered when selecting a block size. Consider the effect of page size on the depth of the trees previously shown in Figure 6.1 A page that holds N records can have N+1 children. The first tree in Figure 6.1 has a height of two levels and holds eight records. This height is ideal because storing the tree's root page in RAM ensures a one-seek retrieval of any record in the tree. Records can be added to the tree by adding more levels. However, this will increase the average number of seeks required for searching. A better plan calls for increasing the block size to accommodate more records. The second tree in Figure 6.1 shows the result of doubling the block size.

Since the CD-ROM is read-only, it is known exactly how many records are going to be put into the tree before it is built. For example, storing 50,000 32-byte records and using a block size of 2K will result in a three-level tree. A two-level tree can be built if a block sizeof 8K is used. It takes longer to read a larger block, but since CD-ROMs can read data at 150K bytes per second, reading an additional 6K bytes takes only 20 milliseconds. This is a small price to pay in return for avoiding an additional 500 millisecond seek. Minimizing the number of seeks is the logical consideration for using large block sizes. However, the CD-ROM's physical features should also be considered in determining what block size to use.

Since the sector size for a CD-ROM is 2K bytes, the smallest block size that should ever be considered is also 2K bytes. This is due to the fact that even if only one byte is needed, 2K bytes will be retrieved. An effective operating system will transfer the data directly into an application program's work area with no intermediate data

movement. So what happens if a program requests only 64 bytes, or some other sector fragment? In this case the operating system cannot assume that the application program has allotted enough space to hold an entire 2048-byte sector. A system buffer must be used to hold the complete sector until the 64 bytes desired can be transferred to the application's work area. Data must be handled or moved twice when anything less than a complete sector is requested. Therefore, in order to avoid unnecessary data movement, a block size that is a multiple of the 2K-byte sector size should be used.

2. **Buffer Usage**

Reading data in multiples of the sector size results in by-passing the system buffers. This blocks the operating system from keeping recently used data in RAM. For example, when a 256-byte record is read, the operating system uses one of its system buffers to hold the sector containing the record. Now another 256-byte record is read in from a different sector. This new sector is placed in a different buffer. The program now calls for a third record which happens to be in the sector which was placed in the first buffer. Therefore, no seek is required for the third record because its sector is already buffered in RAM.

Now suppose instead of reading fragmented records, 2K bytes are read to avoid moving data twice. In this case, system buffers are not used because the data *goes directly to the application* work area. Consequently, a section would be read on top of the first one. In order to benefit from buffering in CD-ROM technology, the decision of how many buffers to provide and how to manage them depends on the nature of the application. If the application searches through tree-structured indexes or works in both directions through a sequence, it can benefit from a large number of buffers. If the application moves sequentially through the data in one direction it will not benefit from buffering at all.

Reference Technology utilizes a general purpose buffering scheme known as Least Recently Used (LRU) replacement. Information in the buffers is retained for user access until buffered data are replaced, according to the least recently accessed protocol. Best performance occurs when the page size is the same as the buffer size and when the number of buffers selected is sufficient to retain the most frequently accessed pages in memory.

Because applications differ, it is impossible to ensure that the most frequently accessed pages will always remain in the buffers. A procedure is needed that will select the minimum number of buffers for maximum performance. Such a procedure would

require that there be at least one more buffer than the number of levels in the tree. Also, there should be at least two buffers for each hash table. The extra buffer per index will hold the data record, while the other holds the index pages. Thus, if two tree indexes with two levels each and two hash indexes were frequently accessed, all with 4096-byte index pages, then $2^3 + 2^2 = 12$ buffers of 4096 bytes each would be the minimum configuration for best performance. (Key, 1986, p. 23)

D. MULTI-VOLUME DISCS

1. Adding Additional Discs

A CD-ROM disc is described, according to the High Sierra standard, as a volume (Standard, 1986, p. 2.5). The standard allows for multi-volume sets of discs, which are of two basic kinds. The first is the type of multi-volume set designed to hold a single massive database that exceeds the capacity of a single disc. The path table and directory structure on each volume of this kind is required to be the same. In this way, the location of any file in the set can be found by reading the directory from any one of the discs. Clearly, it may become necessary to mount a different CD-ROM disc from the set in order to read that file. However, the presence of identical path tables and directories avoids the need to mount disc after disc to find the file of interest.

The second type of multi-volume set of CD-ROMs is necessitated by the need to update files or add new volumes to an existing volume set. If this is the case, the most recent volume's path and directory information must supercede that of all previous volumes. Moreover, the the last volume in the set must be mounted when the system is booted in order to supply the system with the freshest information. By deleting references to a file, or including references to a file in the directory structure of the latest disc in the updating volume set, existing files can be "deleted," "modified," or "replaced." They actually still exist on the earlier discs but since the latest directory no longer points to them, they are no longer available to the system. Although physically present for the life of the CD-ROM, they are logically lost or altered under the present configuration when the new volume is mounted. However, they can be restored if an earlier volume in the set is mounted at system start-up.

2. Extended Attribute Records

CD-ROM file management that is supported within operating systems such as PC-DOS, sees optical disc data as simply a stream of bytes. For other operating environments, extended attribute records (XARs) can provide additional information about the file and its structure. An XAR is an optional attachment to the beginning of

a file, containing extra information about that file. Examples of such additional data include creation and expiration dates, access control, record structure, record attributes, and application-specific information.

One particular use of XARs is to control which version of a file is to be used when there is a multi-volume set of discs containing several versions of a file. This works because the XAR affixed to the last extent of a given file supercedes the XARs affixed to all the other previous extents of that file. If there is no XAR with the last file extent, the XARs with preceding extents are ignored. Thus, by altering the XAR for the final extent of a file, the incidental information about a file is effectively updated when a new CD-ROM is issued.

Another use of XARs is to restrict who may read certain files on a disc. The standard is similar to the VMS "system, owner, group, world" permission design. It should be noted that access restriction only works under those operating systems that recognize it. If someone carries a disc with restricted files to a computer whose operating system, like MS-DOS, does not recognize access protection, the system will read the disc, regardless of the setting of the XAR. Consequently, designing access restriction into a disc must be coupled with a plan to restrict the physical distribution of the discs. (Standard, 1986, p. 2.3)

VIII. CD-ROM APPLICATION SOFTWARE CONSIDERATIONS

A. FILE SYSTEM SUPPORT

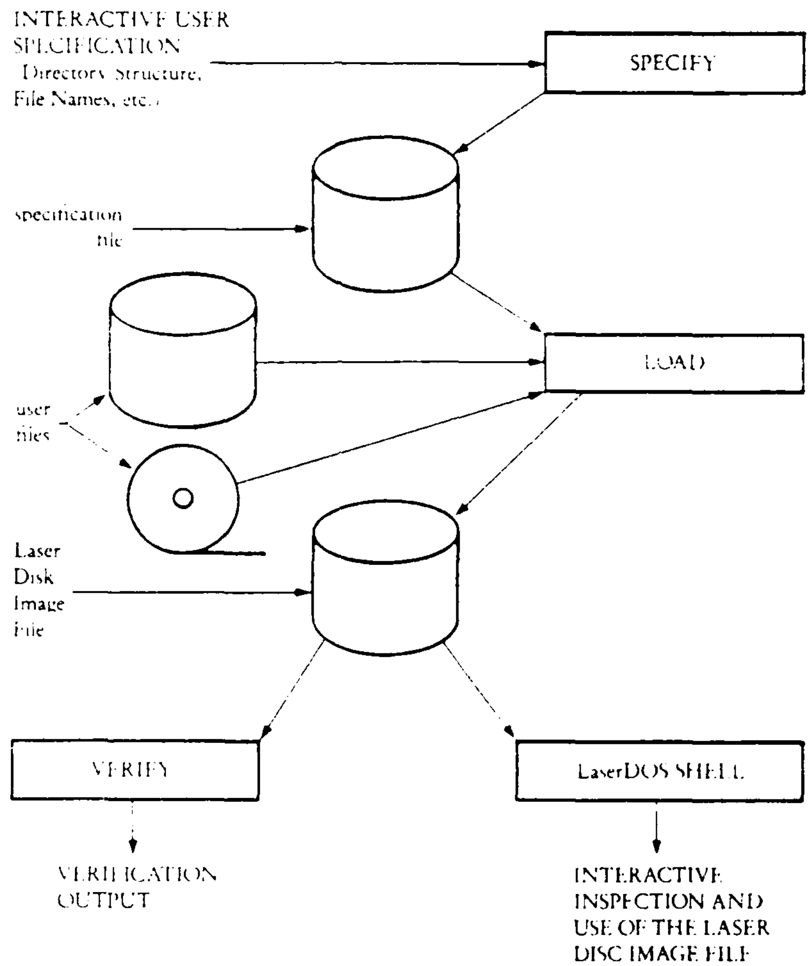
1. Origination Software

Before making a CD-ROM, the files that will appear on the disc must be assembled according to the rules of the logical format. Origination software does this work, providing the writing component of the file system.

At the present time, most origination software runs on minicomputers in batch mode. Figure 8.1 shows the relationship of the four principal components of TMS's LaserDOS origination system. The user begins with a Specify program that provides an interactive shell-like mechanism for creating the directory hierarchy that is to be used on the CD-ROM. During this step the user can indicate which files are to go in which subdirectories. The specification is used as input to a Load process that reads user files from tape and magnetic disk to create a disc image, complete with a volume table of contents and directory structure in the logical format that will be used on the CD-ROM. After loading, the user can run a Verify program that automatically checks the internal consistency and integrity of the disc image. The user can also run a Shell program that exercises the image of the CD-ROM file system interactively, allowing the user to dump out the contents of individual files, copy files to the host operating system, and so on.

2. Destination Software

Destination software is the reading component of the file system. It understands the logical format and uses it to provide access to the CD-ROM files. One way to approach the design of destination software is to create a file manager program containing special function calls that are exclusively for use with the CD-ROM and which bear no relationship to the system calls provided by the host operating system (Zoellick, 1986, p. 125). The advantage of this approach is that the file manager and application programs that use it are not affected by changes in the operating system, thus allowing a higher degree of portability. The main disadvantage is that applications cannot access the CD-ROM through standard system calls which in turn prevents access via high-level language I/O facilities. This makes the CD-ROM less user friendly since familiar language tools and system utilities are unavailable.



Source: CD-ROM The New Papyrus, p. 114.

Figure 8.1 Relationship of Origination Software System Components.

Another design approach involves software such as TMS's LaserDOS and Reference Technology's Standard File Manager, which are implemented for use with MS-DOS (Zoellick, 1986, p. 126). The approach's intent is to cooperate with the host system as much as possible. For example, LaserDOS traps all system calls and determines if the call is CD-ROM related. If it is CD-ROM related it will handle the call itself. If it is not, it simply passes the call on to MS-DOS for completion. The calling software is not smart enough to know the difference. Reference Technology's Standard File Manager works similarly in the TLOCD system. The CD-ROM appears as just another disk drive to the TLOCD user.

B. COMPILER LIMITATIONS

Some compilers used in writing applications that address the file system can in themselves limit the size of files. For example, MS-PASCAL (TM)* (versions 3.1X, 3.20) limits the size of files to eight megabytes. CS6 (TM)* (version 1.2) has the same limit. Lattice C (TM) (version 2.1X) on the other hand is not limited in this way. Reference Technology's Standard File Manager limits itself to file sizes of two Gbytes but the compiler must be capable of producing code that can access a file of this size. PC-DOS has the same two-Gbyte file size limitation as the Standard File Manager if files are accessed through the Standard File Manager "file handling" functions. (Standard, 1986, p. 2.12)

Another potential limitation from compilers is that some restrict the number of files that can be open at one time. For instance, Lattice C (TM) (version 2.1X) has a limit of 20, including the standard input, output, and error files, as well as any hard disk or diskette files. The Standard File Manager for CD-ROM systems allows up to 200 files to be open simultaneously.

C. PC-DOS ADAPTATION

One of the more frustrating things about using CD-ROM with IBM PCs is the limitation placed on the size of a logical disc volume by the PC-DOS operating system. It is only 32 megabytes--a mere thimble full compared to the 540 megabytes typically available on a single CD-ROM. Fortunately, there are several ways to sidestep this limitation. One relatively easy way is to surrender to PC-DOS and break the disc into 32-megabyte partitions.

However, the most powerful method to get around the size limitation involves a new interrupt handler. It may also be necessary for the file-management system as well

as the directory depending on how the particular system is set up. By trapping the operating system interrupt, the interrupt handler can intercept calls intended for the CD-ROM while other calls are simply passed through. Once intercepted, the CD-ROM calls can be treated differently, still maintaining system transparency to the user.

The difficulty arises when the interrupt handler must also support every disc call in exactly the same way as PC-DOS supports them. Those calls include functions that open files, read from files, check for remaining disc space, and so forth. Supporting all of those functions necessitates a tremendous amount of code generation.

IX. ANALYSIS AND DISCUSSION

A. SHIPBOARD USE OF CD-ROM

1. Departmental Applications

There are many applications for CD-ROM systems on board U.S. Navy vessels. Such applications will decrease the ship's weight (by eliminating paper storage media) and make more space available. The advantages, and disadvantages and possible problem solutions are addressed.

The Navigation department should store its hundreds of charts on CD-ROM and eliminate a majority of its bulky chart cabinets. The system would store the charts in ascending order according to chart number and would also provide a cross-reference index for user assistance. The system would prompt the user to enter the number of the chart he wishes to see and then display that chart on the monitor. However, there must be a system on board for reproducing these charts into a paper medium so that corrections, courses, fixes, and coordinates can still be plotted. The technology needed to reproduce NOAA charts in various scales is now available from LaserPlot, Inc.(Belanger, 1987, p. 13).

The Operations department should use CD-ROM to hold its classified publications. Security will be better because there will be fewer classified materials to be monitored. Confidential material would be kept on one CD-ROM, Secret material on another, and Top Secret material on still another. However, in environments such as MS-DOS, security becomes breeched when a person with the "need to know" about a certain topic has access to all other classified information that resides on the disc he happens to be reading. In that case, software would have to be developed in which the ship's CMS custodian would control a "read denial" lock for each classified file. The operating system would not relinquish control to the CD-ROM file manager without checking the lock status. The lock could only be set or reset according to a program executed by the CMS custodian. No file could be opened and read without the custodian's knowledge and approval. An individual would sign for the CD-ROM and the CMS custodian would release the locks on those files that the user is qualified to view. Upon the return of the classified disc the lock would be reset. Another particularly helpful CD-ROM application in the Operations area involves "signal

breaking" or tactical communications. Such an application should be written to search through tactical publications such as NWPs and AWPs and break coded signals, thereby ensuring timeliness and accuracy in situations that can be and often are critical. The tactical officer would key in the coded signal phrase and the system would search its database for that particular sequence of words. The results would be displayed on monitors located on the ship's bridge and in CIC.

The Engineering department maintains a vast number of operating manuals, technical manuals, repair manuals, and schematics. The transfer of these from paper to CD-ROM would certainly reduce weight and increase available departmental space. The engineers would also have access to many more manuals, blueprints, and technical publications not normally carried on board. But how is a repairman going to get a repair manual to the scene of repair? Must he go to a CD-ROM reader and print out the applicable pages? The answer is a qualified yes. A repairman will usually have to go to a centralized location to check out a manual. Disc readers and printers should be placed in these strategic locations in order to minimize the inconvenience. In certain circumstances, with the use of some advanced technology, a print-out may not be necessary.

The Supply department should use CD-ROM to store its wide variety of catalogs, parts lists, and various other publications. Cookbooks and recipes would no longer be lost or misplaced. All of these potential uses would be complemented by the CD-ROM's ability to store visual images. The supply clerks can see exactly what they are ordering and thereby reduce errors that often result from making assumptions or guessing about item uncertainties. Moreover, CD-ROMs already contain the Navy Management Data List (NMDL) and Parts I and II of the Master Repair Items Listing (MRIL) which is distributed by the Navy Publications and Printing Service. NAVSUP also sponsored the TLOCD project done here at the Naval Postgraduate School.

The Administration department would no longer have to print and distribute copies of Navy-wide regulations and instructions throughout the ship. The drawback here is a lack of shipboard portability. For example, the person desiring the information must be in the immediate vicinity of a CD-ROM disc reader. He cannot go to his stateroom, relax, and thumb through the newest instruction or regulation--unless, of course, there happens to be a CD-ROM disc reader in his stateroom. This scenario is not unrealistic. Considering that the total cost of a disc reader, monitor,

keyboard, and printer can be held under $1,500.00, it is feasible that such a system could be placed in nearly all the spaces on board the ship. Costs could be reduced further if a networking system were implemented and public terminals made available to the crew. One possible networking scheme would involve a modem to modem machine interface using the ship's telephone lines. However, this method might interrupt routine shipboard communications by tying up the phone lines. A better solution would involve the development of a local area network (LAN) which would allow as many users as there were system hook-ups. Each compartment would be wired so that portable terminals could be supported. The structure would be relatively simple for such a system and could be supported by a common network topology such as a ring. The decision to implement a LAN or to pursue a certain network topology across a particular class of ships should be made by NAVSEA based upon fleet managerial requirements determined by individual ship needs.

2. CD-ROM Impact on the Paperless Ship

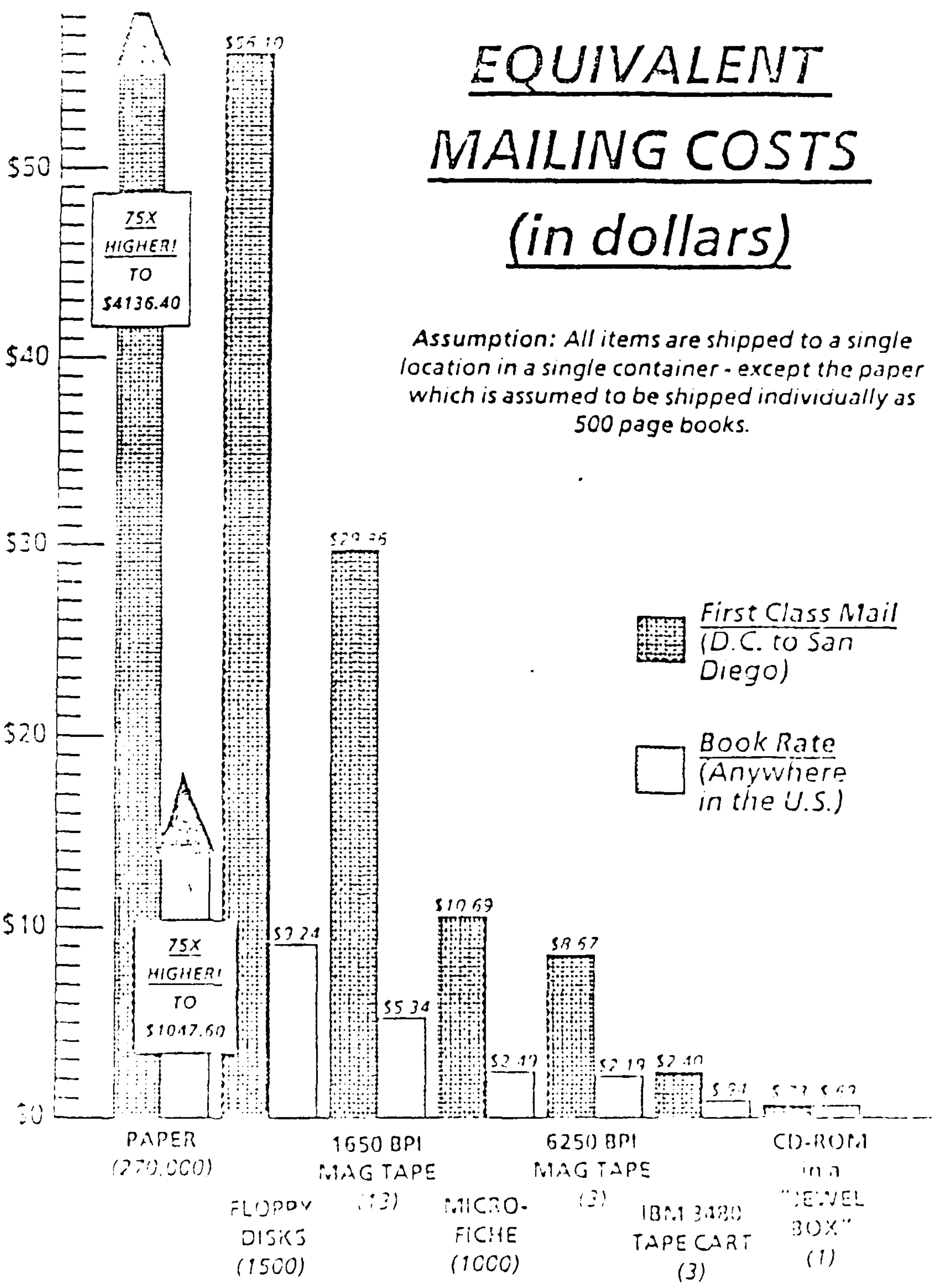
Every officer and petty officer aboard every Navy ship has at one time or another become frustrated by the unending flow of required paperwork and the plethora of information in technical manuals and documents that must be available, read, and studied. Cumulatively, their weight is in tons. VADM J. Metcalf III states,

> "I find it mind-boggling. We do not shoot paper at the enemy. We do not train sailors to be registrars and correctors of publications. I want those guys worried about fighting, not worrying about keeping up the publications."

The admiral has launched an initiative to create a "paperless" ship by 1990 as a first step toward driving paper from the entire fleet. The first ship would be a frigate, he said, that will probably be equipped with different types of electronic information systems. (Metcalf, 1987, p. 35)

CD-ROM technology is only a piece of the puzzle when it comes to putting together such a system. One must consider the feasibility of making CD-ROM disc readers accessible to all departmental and divisional offices as well as in CIC, DCC, the Bridge, engineering spaces, and staterooms. The initial cost would be considerable but would be offset in a short while by the reduction in mailing costs of optical discs as opposed to paper. See Figure 9.1 for a comparison between mailing costs of CD-ROM and other storage media.

Keyboards, monitors, printers, and disc readers must be kept in a relatively cool environment in order to reduce downtime and maintain operational readiness.



Source: Lind Thesis, p. 25.

Figure 9.1 CD-ROM Mailing Comparison.

Many ships are not currently capable of producing such an environment with any consistency--especially in humid climates such as the Persian Gulf, Indian Ocean, or Caribbean Sea. The newer ship classes, however, should not experience as many problems because of additional electronics needs being addressed in the ships' original design. Furthermore, the loss of ship's power could prevent timely access to important data. In that case, it would be necessary that a paper copy of such data be stored on board. An alternative solution would be to require each major user to have his own back-up power source such as an UPS (uninterruptible power supply) which runs off its own battery pack until a diesel or gas engine is started and begins to produce the power source. It is possible to have an UPS for the entire shipboard computer system but it would require larger battery packs. The decision on how to employ UPS is again strictly a managerial one based on individual ship characteristics and goals.

Another problem that surfaces involves applications such as personnel or disbursing transactions that require constant change or update. Write Once Read Many (WORM) optical technology may be the solution in these cases. Other emerging technology that may be available in the near future includes erasable optical discs which function in much the same way as a standard floppy disk. The goal of a paperless ship is certainly obtainable if CD-ROM is used in conjunction with other eletronic media such as WORM. However, in order for this to happen, ships must maintain a cool operating environment, shipboard portability issues must be resolved, and the use of additional electronic data storage methods to compensate for the CD-ROM's weaknesses must be available and cost effective.

B. CD-ROM FOR SHORE FACILITIES

1. Database Design

The use of CD-ROM at U.S. Navy shore facilities must be tailored to fit the needs of the particular command. The storage and retrieval of massive amounts of historical data is the primary consideration for implementing a CD-ROM system such as the TLOCD system at NSC Oakland. Database design demands considerable attention from facilities wishing to effectively capitalize on the read-only nature of CD-ROM technology. Of particular concern is the format of the database. CD-ROM databases may consist of a number of files--each file consisting of similar records having the same logical format. Since a database from a CD-ROM perspective is a collection of similar files concatenated together, a single optical disc may contain many distinct databases of different file types. In this case, the TLOCD system actually

involves three distinct databases--one each for the transaction files, closing balance files, and audit trail files.

When designing a database, attempts should be made to maximize the system's storage allocation potential. This consideration was neglected in the TLOCD design. Consequently, many of the records in each of its three databases contain data common to records in the other two databases. For example, the National Item Identification Number (NIIN) and date fields are found in all three record types of the TLOCD system. This data redundancy across databases should be avoided whenever possible in order to achieve a higher level of storage efficiency.

Care should be taken not to merge separate entities such as the TLOCD databases in an attempt to delete redundant information. Such an attempt could lead to wasted space, continued data redundancy, and unwanted loss of valuable data. Note Figure 4.2 in which three fictitious file tables of the TLOCD system are merged into a single table made up of tuples that represent data records. Notice that there are no entries in some of the record fields. The space must still be maintained and is virtually wasted. Now notice the data redundancy among the record fields. Furthermore, if a record were ever to be damaged or destroyed the audit trail data for that date would be lost, resulting in an inaccurate historical account of inventory items. That is the reason why multiple entities should not be routinely merged into a single table to reduce redundancy when designing a database for a particular system.

2. Cost Effectiveness

Businesses today are constantly in search of managerial tools and manufacturing procedures that reduce overhead and still maintain product reliability. The U.S. Navy is no different. There are two specific areas in CD-ROM projects such as TLOCD where costs could be trimmed. The first such area deals with indexing. The total cost for preparing and creating the TLOCD indexes exceeded $9,000 (Lind, 1986, p. 59). The Navy may benefit from providing its own indexing and utilizing $9,000 in cost savings elsewhere. Any Navy facility with sufficient computer hardware can create the indexes required for CD-ROM manufacturing. In fact, there are hardware and software units now available that can perform all stages of CD-ROM production through the premastering stage. The "CD Publisher" from VideoTools is one such product. However, it would be a simple task to assign the job of indexing to a mini-computer which could grind out the results in batch mode. The main concern would be in deciding the type of index structure to use for the particular application in order to

TRANSACTION TABLE

NIIN	DATE	NAME	NSN	TRANSACTION DATA
123456789	7260	BOLT	889	+ 5000
345678901	7290	NUT	732	- 1200
567890123	7290	WASHER	654	- 2500
789012345	7300	NAIL	098	+ 8000

CLOSING BALANCE TABLE

NIIN	DATE	NAME	NSN	BALANCE DATA
234567890	7260	SCREW	339	9000
345678901	7270	NUT	732	25000
567890123	7290	WASHER	654	950
678901234	7290	FASTENER	492	4317

AUDIT TRAIL TABLE

NIIN	DATE	NAME	NSN	AUDIT TRAIL DATA
123456789	7260	BOLT	889	MOVED 3000 TO AB
345678901	7270	NUT	732	MOVED 2000 TO JK
456789012	7280	SPACER	111	LOANED 5 TO NAV
678901234	7290	FASTENER	492	MOVED 95 TO JAX

MERGED TABLE

NIIN	DATE	NSN	NAME	TRANSACTION DATA	CLOSING BALANCE	AUDIT TRAIL
123456789	7260	889	BOLT	+5000		MOVED 3000 TO AB
234567890	7260	339	SCREW		9000	
345678901	7270	732	NUT		25000	MOVED 2000 TO JK
345678901	7280	732	NUT	-1200		
456789012	7280	111	SPACER			LOANED 5 TO NAV
567890123	7290	654	WASHER		950	
567890123	7290	654	WASHER	-2500		
678901234	7290	492	FASTENER		4317	MOVED 85 TO JA
789012345	7300	098	NAIL	+8000		

Figure 9.2 Merging of Multiple Entities.

maximize performance. Therefore, some knowledge of CD-ROM indexing would be essential.

The second area in which costs could be trimmed involves application software. The TLOCD application specific software was created at a cost of about $4,500. Qualified Navy personnel can create programs to access the TLOCD database using the library of C language functions already resident in the *Key Record Manager*. Programmers having experience in a high level language should be able to develop sufficient C programming skills within a short time and then produce programs for TLOCD and other naval applications. Granted, it is necessary to purchase software such as the *Key Record Manager* to interface with the CD-ROM file management system or else write an independent interface. However, that might not seem very prudent since the time and cost to develop and debug such an interface would certainly prove more costly than an already proven product such as *Key Record Manager* which has been sold commercially for under $200. Furthermore, such a task would require a great deal of systems programming in a language such as C at a time when DoD has declared ADA to be the primary language to be utilized in future military projects. Since most CD-ROM access software on the market today is C-language oriented, the Navy should direct research toward developing ADA programs to drive CD-ROM applications. There are indications from the CD-ROM industry that ADA interfaces will be available on the consumer market within a few months. An alternative to this approach would be an interface written to accommodate any compiled code recognizable in the operating system extensions, therefore allowing several different compiled languages to access it.

C. TLOCD PROTOTYPE IMPROVEMENT

1. Proposed System Modification

As stated previously, the current TLOCD system accesses and searches three distinct databases in order to obtain transaction, closing balance, and audit trail information for inventory item inquiries. The system should be modified by extracting the redundant data from the databases without destroying the separate entities or relations among the three file types. This could be accomplished by restructuring the files. Duplicate data would be removed from the three files and placed in a separate table or "NIIN file" which is then linked to the other tables via multiple pointers from the NIIN table or via a chaining mechanism from one table to the next. Although the number of tables is now increased by one, such an arrangement does not imply

inefficiency. The data storage capacity is increased and the tables remain in as separate entities to be used for other purposes. This new structure would provide three TLOCD files without duplicate data in such a way that the separate entities associated with the TLOCD files each have attributes that apply to that particular entity. Therefore, the storage requirement is reduced without removing the idea of separate entities--which is a requirement for TLOCD system control.

2. **Functional Design Issues**

In designing a system such as TLOCD there are three issues of primary concern: database access, data search, and data retrieval. These criteria will now be discussed in relationship with the proposed TLOCD modifications.

Accessing the TLOCD database involves locating and "opening" its index and data files. The access function must search the CD-ROM database directory for the database name provided by the user or the user's program. The address of a File Control Block (FCB) is acquired from the database directory. The FCB will contain a pointer to a list of the key record indexes used for searching the database. It also will contain a pointer to the beginning address of the actual data on the CD-ROM. This "double-pointer" configuration allows the system to search a specified index for a key record value and acquire the relative address of the record within the data file. The pointer within the data file is then utilized to locate the record. In this way the integrity of the pointers can be maintained and subsequent searches can be conducted relative to the current pointer positions. Such an access function requires two parameters--the database name as an input parameter and the database address as an output parameter.

The primary objective of the TLOCD system is to obtain historical data about a particular NIIN for a specified date. Therefore, the most important fields within the data records are the NIIN and date fields. The NIIN is used to generate a key record index. The date field is not used as an index generator. It would not provide a practical key record index since there could be possibly hundreds or thousands of transactions conducted on that particular date. Other fields that would generate adequate key record indexes include the National Stock Number (NSN) and the product noun name. However, since the TLOCD system users deal primarily with the NIIN and seldom have the need for additional identifiers, no other key indexes would be utilized on a regular basis.

Normally, indexes are numbered sequentially and the user is queried as to which index he desires to search. However, since only the NIIN index is to be created for the TLOCD system modification, no query is needed and the NIIN index is selected by default. The user is prompted to enter the NIIN and the date if it is known or desired. The NIIN is located in the index via a balanced tree search. A pointer is then followed to a list of date records containing the dates on which the NIIN was transacted and the offsets of their associated NIIN records within the file. The dates are listed in ascending numerical order according to their Julian equivalents. The NIIN record offset is retrieved, record address computed, and the pointer is moved to the desired record of the NIIN file. Input parameters for such a search function include: (1) the database address, (2) the index to be searched, (3) the NIIN, and (4) the date. The function will return the record offset in relation to the NIIN file origin. If no date is specified, the function will return the offset for the earliest recorded transaction for the specified NIIN. See Figure 9.3 for an illustrative example.

Once the record is located in the data file its contents must be retrieved and displayed for the user. There are various methods that can be used to achieve the task. One such method involves the use of a function similar to the "scan" function found in the C programming language. In such a technique, the record is treated as a string of bytes and the string is "scanned" or read into a buffer. The contents of the buffer are then displayed on the screen. In order to make any sense of the data, other functions must be called upon to format the record string into a readable medium. The record size must be known so the scan function can determine how many bytes to transfer into the buffer. This poses no problem for the TLOCD system since its records are of fixed length. However, for variable length records, the scan function would have to be designed to look for a length field at the beginning of each record--or else receive the information from the search function. Data retrieval can be similarly executed by string manipulation functions commonly found in such programming languages as Pascal and ADA. Retrieval programs written in C warrant more consideration due to the language's powerful screen formatting functions.

3. Other Issues

No system design can afford to ignore the needs and desires of its user environment. Systems that are not user friendly seldom make an impact in the market place. Such essential TLOCD user response has indicated dissatisfaction with the "page up" and "page down" functions that permit them to move forward or backward

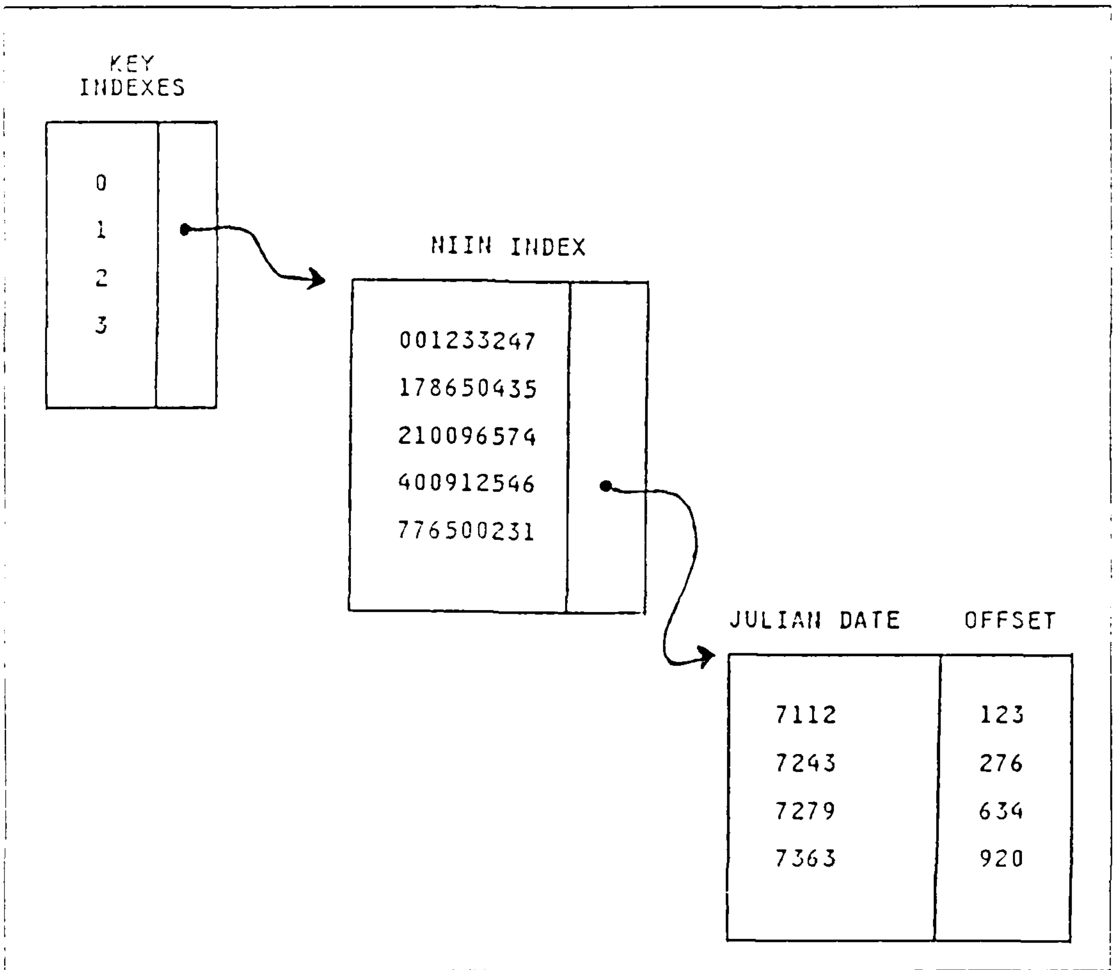


Figure 9.3 Search for Specific NIIN.

within the data file only one record at a time. They would benefit from a scroll function which would allow them to move forward or backward within the file any number of records. Such a function would not be hard to implement and would add flexibility for users. The user would provide an integer (positive or negative) input for the number of records he wishes to scroll over. Since the records are of fixed length, such a function could readily compute the new position of the record in the data file and then reposition the pointer to that location. The function would require three input parameters: (1) current pointer position, (2) record length, and (3) number of records to scroll. It would pass the new record location as an output parameter. An attempt to scroll past the beginning or end of the data file would result in retrieval of the first or last record in the file.

Another issue to be concerned with is the arrangement of data on the terminal screen. The current TLOCD screen interface displays a transaction record for a specific NIIN and then queries the user as to whether he wants to view a closing balance or audit trail record for the NIIN. Therefore, the user is aware that he must deal with three separate groups of files. The user has no need to know such information and the system should make it transparent to him. Furthermore, the screen interface should display data from across all three TLOCD relations upon each NIIN inquiry. The result would be a fuller screen with multiple records being used to provide transaction, closing balance, and audit trail data about the NIIN. The need no longer exists to prompt the user after each NIIN search to query the user about closing balance or audit trail data.

The design of a user-friendly interface to a system is a complex one and goes beyond the scope of this thesis. The above examples serve to illustrate that these issues must be carefully analyzed to provide user satisfaction.

X. CONCLUSIONS AND RECOMMENDATIONS

The U.S. Navy is constantly exploring, experimenting, and seeking new technologies in order to maintain a tactical advantage over its adversaries. CD-ROM technology warrants immediate attention and funding for implementation and applications development.

CD-ROM applications provide a potentially valuable commodity to the U.S. Navy at shore facilities and on board ships at sea. The product is already proven and the financial risks are minimal. Major shore facilities should proceed and adopt plans to convert their permanent and archival databases to CD-ROM applications such as the TLOCD system. The technology is available and is already starting to earn a significant niche in the electronic data processing industry. Although an implementation reflecting the proposed TLOCD modifications presented in the previous chapter cannot be carried out within the scope and time frame of this thesis, it can be determined from the information presented that such an implementation is plausible and doable within U.S. Navy environments.

CD-ROM is the catalyst that will eventually lead to the first paperless ship. Its use in conjunction with other developing electronic technology such as WORM makes the goal reachable. The Navy should designate a ship to function as a prototype for CD-ROM conversion. The prototype must apply sound database design principles such as those emphasized in this study in order to produce efficient and effective performance. It must also address the functionality of the user interfaces designed for each specific application on an independent basis. If these guidelines are followed, the CD-ROM applications will produce immediate cost savings and increase efficiency and operational readiness by providing faster access to critical data. If current research and development cannot economically produce a feasible optical storage solution (such as WORM or erasable discs) for constantly changing data, then the chances for a "paperless" ship in the near future are greatly reduced. Regardless of that outcome, CD-ROM will remain reliable and cost-effective for shipboard use providing proper analysis is conducted prior to system integration.

LIST OF REFERENCES

Abeytunga, P.K., "Canadian C.C.O.H.S. Releases CD-ROM Database," *Video Computing*, February 1987.

Belanger, Michael, "LaserPlot Sails Into CD-ROM Market With Navigation System," *Video Computing*, March 1987.

Colvin, Gregory, "Database Retrieval and Indexing," *CD ROM Optical Publishing*, Microsoft Press, 1987.

"Contract Awards," *Government Computer News*, Vol. 5, No. 20, 10 October 1986.

Cooke, D.F., "Maps, Optical Discs, and Vehicle Navigation," *CD ROM The New Papyrus*, Microsoft Press, 1986.

DeFray, Jeff, "CD-ROM: A Curiosity No Longer," *CD-ROM Review*, premiere issue, [illegible].

Dixon, B.N., "The Grolier Electronic Encyclopedia," *CD-ROM Review*, Vol. 2, No. 1, March April 1987.

Duiade, J.R., "The Application Interface of Optical Drives," *Byte*, Vol. 11, No. 5, May 1986.

Einberger, John, "CD ROM Characteristics," *CD ROM Optical Publishing*, Microsoft Press, 1987.

Gates, W.H., "Foreword," *CD ROM The New Papyrus*, Microsoft Press, 1986.

Hensel, D., "Data Publishing on Optical Discs," *CD ROM The New Papyrus*, Microsoft Press, 1986.

Huntting, Stan, "Open Wide and Say Dataaa," *CD ROM The New Papyrus*, Microsoft Press, 1986.

Key Record Manager, version 2.00, Reference Technology, 1986.

Kuhn, Carolyn, "The Visual Dictionary," *CD-ROM Applications Forum*, No. 1, May 1987.

Lind, D.J., *Optical Laser Technology, Specifically CD-ROM and Its Application to the Storage and Retrieval of Information*, M.S. Thesis, Naval Postgraduate School, Monterey, California, March 1987.

Melin, Nancy, "The New Alexandria: CD ROM in the Library," *CD ROM The New Papyrus*, Microsoft Press, 1986.

Metcalf III, VADM J., "Surface SITREP," *Surface Warfare*, Vol. 12, No. 4, Jul Aug 1987.

Moore, Steve, "The Mass Storage Squeeze," *Datamation*, Vol. 30, No. 16, 01 October 1984.

Standard File Manager, version 1.00, Reference Technology, 1986.

[illegible], Rick, "Graphics Show Has a Little of Everything," *Government Computer News*, Vol. 5, No. 23, 21 November 1986.

Zoellick, B., "File System Support For CD ROM," *CD ROM The New Papyrus*, Microsoft Press, 1986.

Zoellick, Bill, "CD-ROM Software Development," *Byte*, Vol. 11, No. 5, May 1986.

BIBLIOGRAPHY

Abeytunga, P.K., "Canadian C.C.O.H.S. Releases CD-ROM Database," *Video Computing*, February 1987.

Belanger, Michael, "LaserPlot Sails Into CD-ROM Market With Navigation System," *Video Computing*, March 1987.

Colvin, Gregory, "Database Retrieval and Indexing," *CD ROM Optical Publishing*, Microsoft Press, 1987.

"Contract Awards," *Government Computer News*, Vol. 5, No. 20, 10 October 1986.

Cooke, D.F., "Maps, Optical Discs, and Vehicle Navigation," *CD ROM The New Papyrus*, Microsoft Press, 1986.

DeTray, Jeff, "CD-ROM: A Curiosity No Longer," *CD-ROM Review*, premiere issue, 1986.

Dixon, B.N., "The Grolier Electronic Encyclopedia," *CD-ROM Review*, Vol. 2, No. 1, March April 1987.

Dulude, J.R., "The Application Interface of Optical Drives," *Byte*, Vol. 11, No. 5, May 1986.

Einberger, John, "CD ROM Characteristics," *CD ROM Optical Publishing*, Microsoft Press, 1987.

Gates, W.H., "Foreword," *CD ROM The New Papyrus*, Microsoft Press, 1986.

Hensel, D., "Data Publishing on Optical Discs," *CD ROM The New Papyrus*, Microsoft Press, 1986.

Huntting, Stan, "Open Wide and Say Dataaa," *CD ROM The New Papyrus*, Microsoft Press, 1986.

Key Record Manager, version 2.00, Reference Technology, 1986.

Janus, Susan, "The Mass-Storage Boom," *PC WEEK*, Vol. 3, No. 24, 17 June 1986.

Kuhn, Carolyn, "The Visual Dictionary," *CD-ROM Applications Forum*, No. 1, May 1987.

Lind, D.J., *Optical Laser Technology, Specifically CD-ROM and Its Application to the Storage and Retrieval of Information*, M.S. Thesis, Naval Postgraduate School, Monterey, California, March 1987.

Melin, Nancy, "The New Alexandria: CD ROM in the Library," *CD ROM The New Papyrus*, Microsoft Press, 1986.

Meng, Brita, "Optical Disks Slip on Compatibility," *Digital Design*, January 1986.

Metcalf III, VADM J., "Surface SITREP," *Surface Warfare*, Vol. 12, No. 4, Jul Aug 1987.

Moore, Steve, "The Mass Storage Squeeze," *Datamation*, Vol. 30, No. 16, 01 October 1984.

Standard File Manager, version 1.00, Reference Technology, 1986.

Vizachero, Rick, "Graphics Show Has a Little of Everything," *Government Computer News*, Vol. 5, No. 23, 21 November 1986.

Vizachero, Rick. "New Retrieval System Uses Laser Disks." *Government Computer News*, Vol. 5, No. 20, 10 October 1986.

Vizachero, Rick. "Optical Storage Gets Agencies' Attention." *Government Computer News*, Vol. 5, No. 24, 5 December 1986.

Whieldon, David. "Look at Optical Storage." *Computer Decisions*, 2 December 1986.

Zoellick, Bill. "File System Support For CD ROM." *CD ROM The New Papyrus*. Microsoft Press, 1986.

Zoellick, Bill. "CD-ROM Software Development." *Byte*. Vol. 11, No. 5, May 1986.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chief of Naval Operations Director, Information Systems (OP-945) Navy Department Washington, D.C. 20350-2000	1
4.	B.A. Frew Administrative Sciences Department Code 54FW Naval Postgraduate School Monterey, CA 93943	3
5.	Dr. Daniel Davis Computer Science Department Code 52DV Naval Postgraduate School Monterey, CA 93943	1
6.	Commanding Officer Naval Supply Center Code 61 Oakland, CA 94625-5000	1
7.	Director Naval Data Automation Command Washington Navy Yard Washington, D.C. 20374-1662	1
8.	LT J.S. Johnson 213 Knob Road Lebanon, TN 37087	2
9.	Commander, Naval Supply Systems Command Code 064 Department of the Navy Washington, D.C. 20350	1

10. United States Military Academy 1
Department of Geography and Computer Science
West Point, NY 10996-1695

11. Commanding Officer 1
Naval Data Automation Command (Code 30)
Attn: LCDR David J. Lind
Washington Navy Yard
Washington, D.C. 20374-1662

12. Commanding Officer 1
Navy Fleet Material Support Office
P.O. Box 2010
Code 9513
Mechanicsburg, PA 17055

END

DATED

FILM

8—88

DTIC